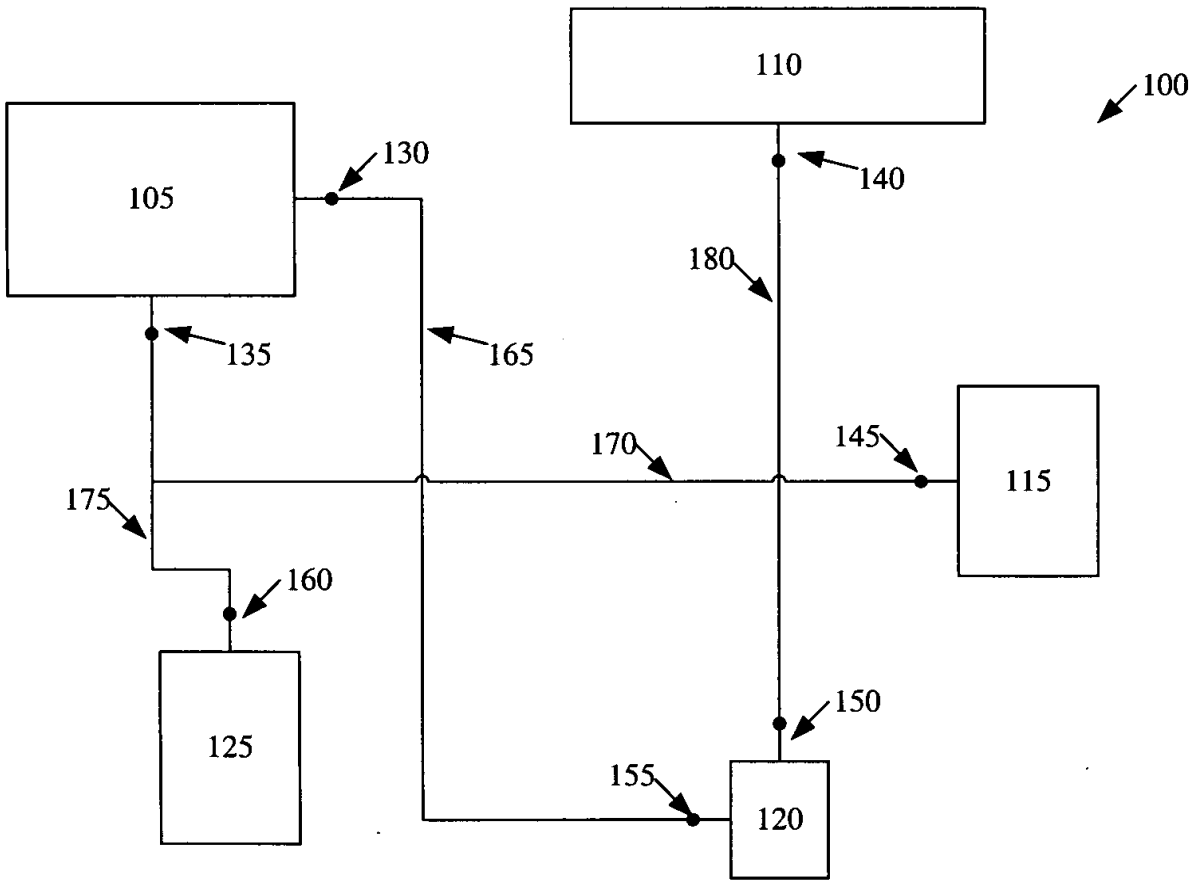

Figure 1

10030618 - 041702

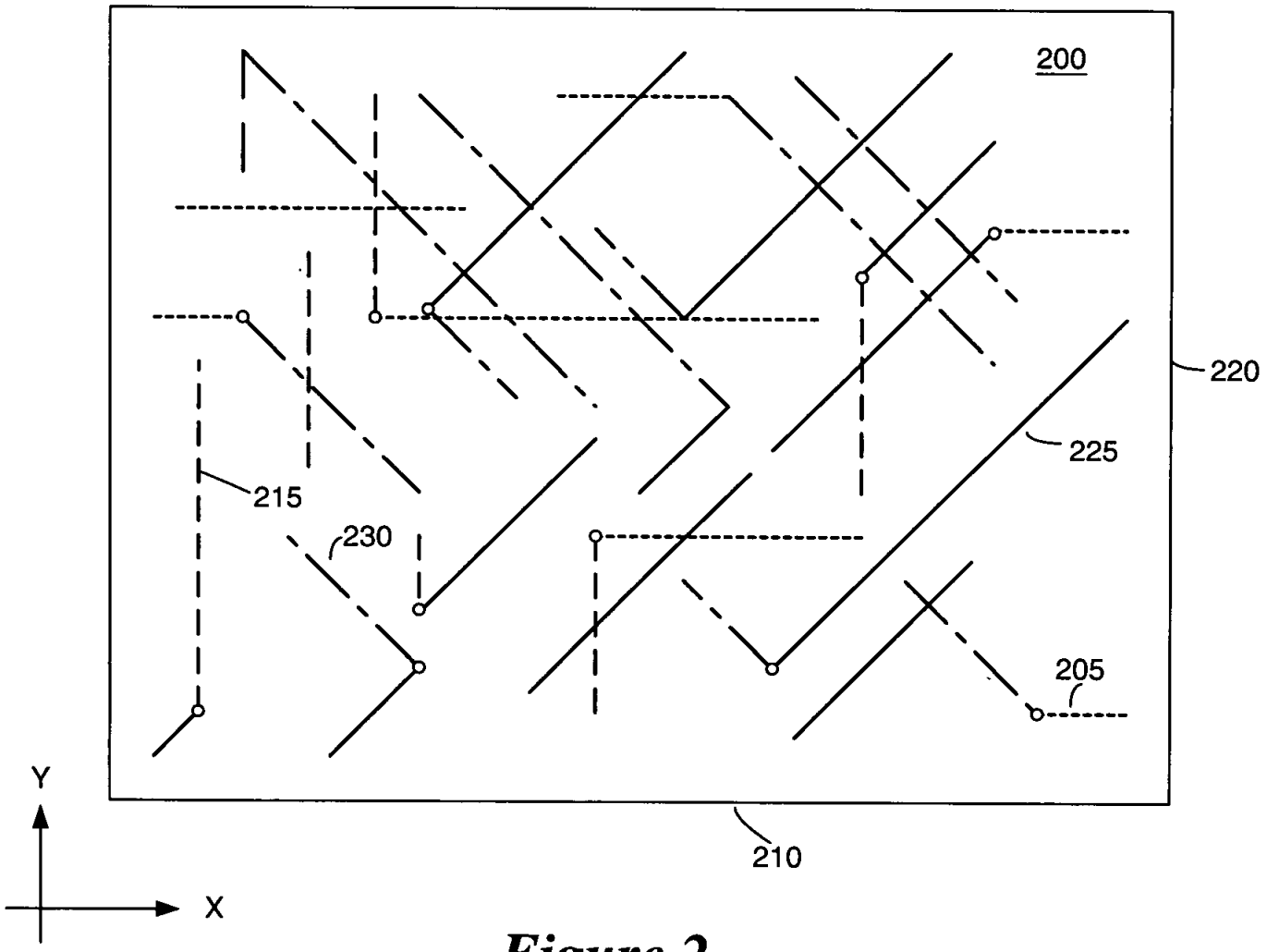

Figure 2

20041702.041810.1002956

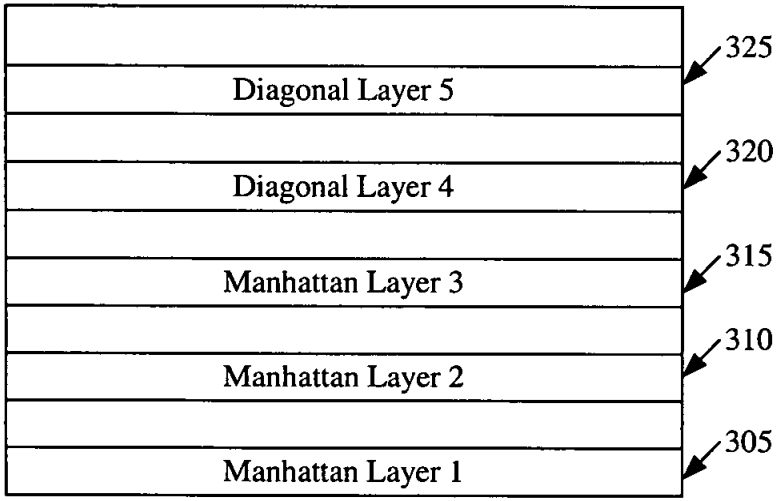
Diagonal Layer 5
Diagonal Layer 4
Manhattan Layer 3
Manhattan Layer 2
Manhattan Layer 1
325
320
315
310
305

Figure 3

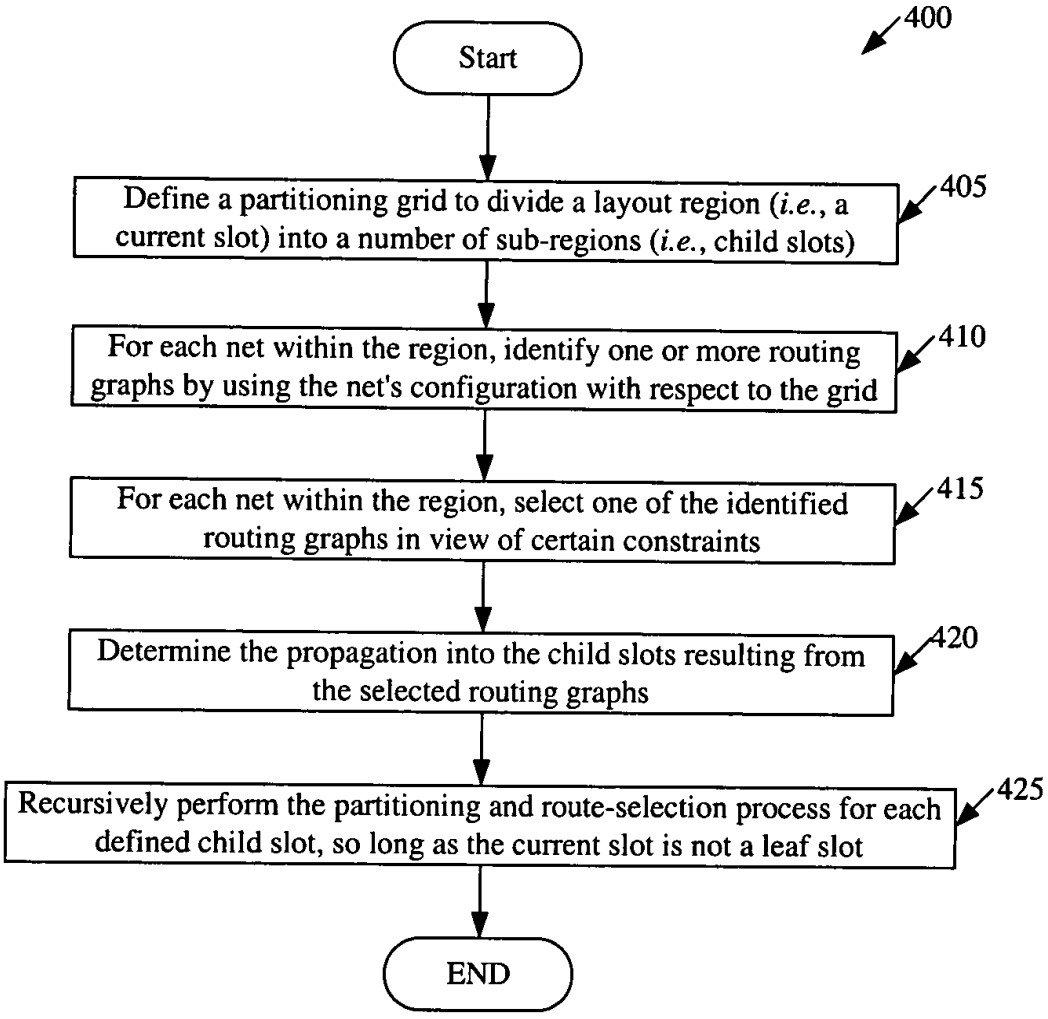
400
Start
Define a partitioning grid to divide a layout region (i.e., a current slot) into a number of sub-regions (i.e., child slots)
405
For each net within the region, identify one or more routing graphs by using the net's configuration with respect to the grid
410
For each net within the region, select one of the identified routing graphs in view of certain constraints
415
Determine the propagation into the child slots resulting from the selected routing graphs
420
Recursively perform the partitioning and route-selection process for each defined child slot, so long as the current slot is not a leaf slot
425
END

Figure 4

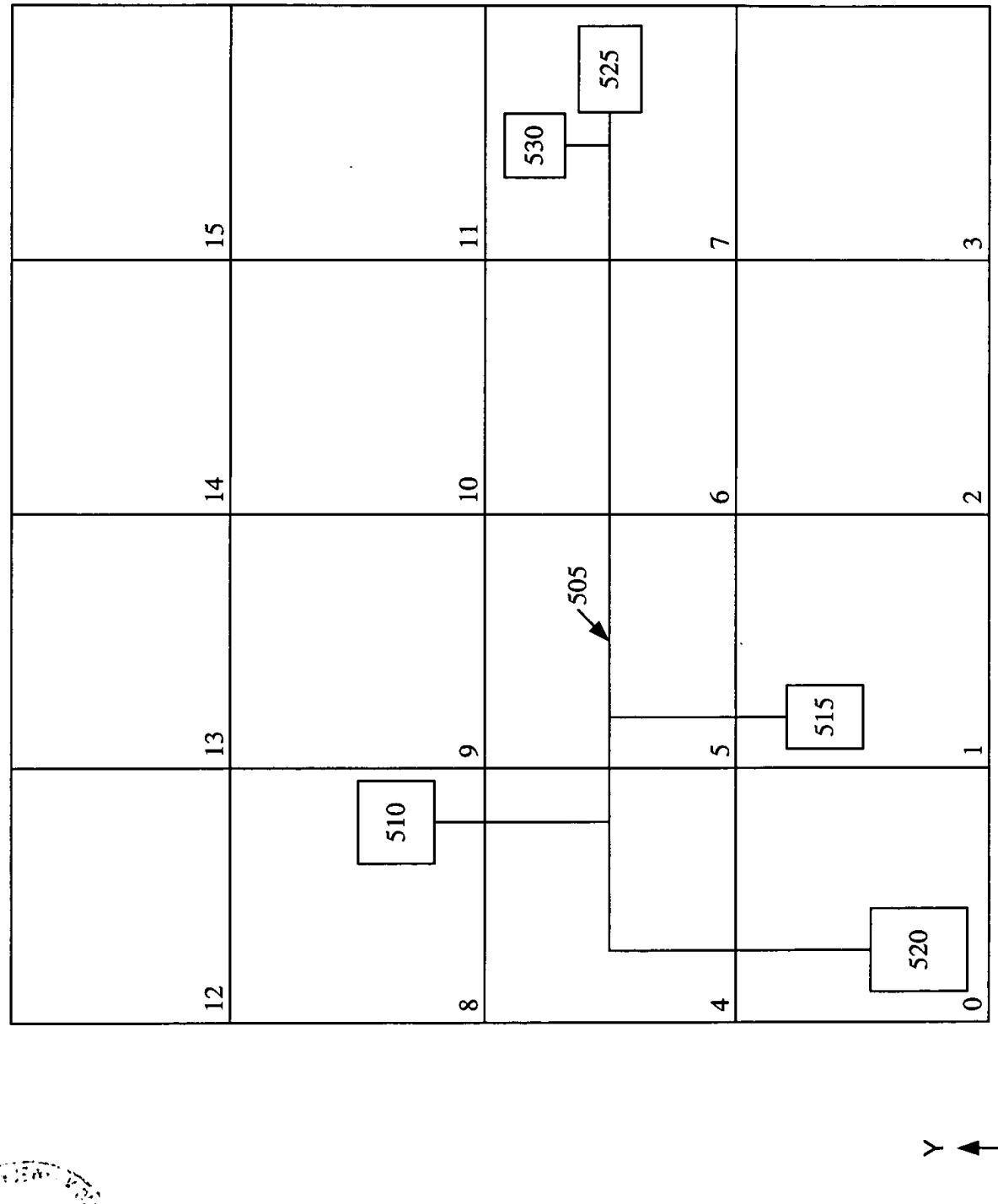

Figure 5

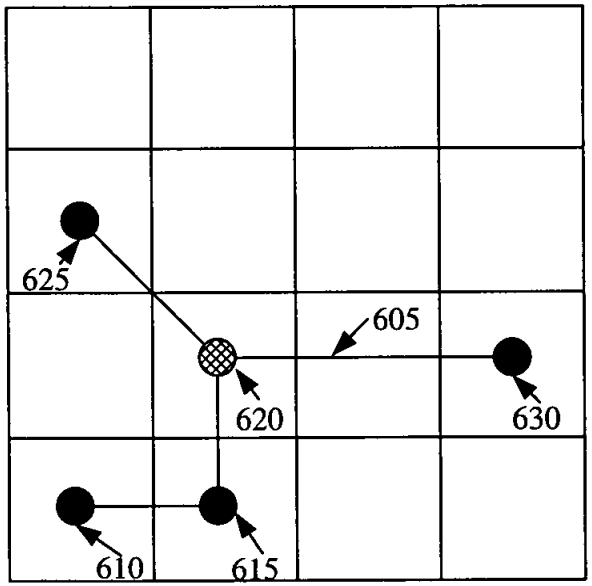

Figure 6

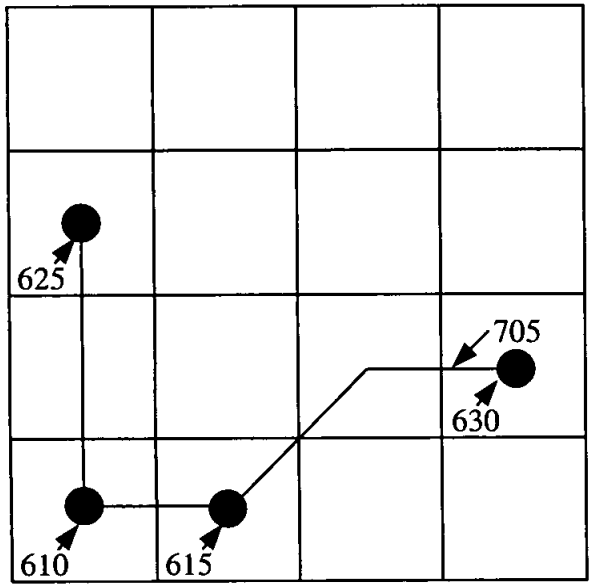

Figure 7

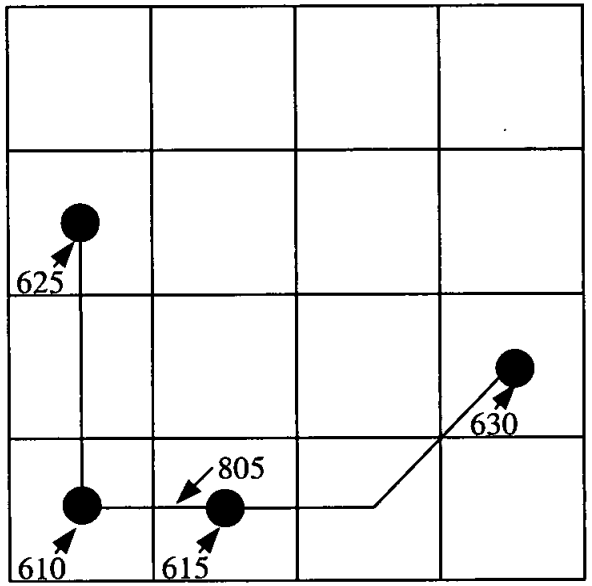

Figure 8

10025618 - 041702

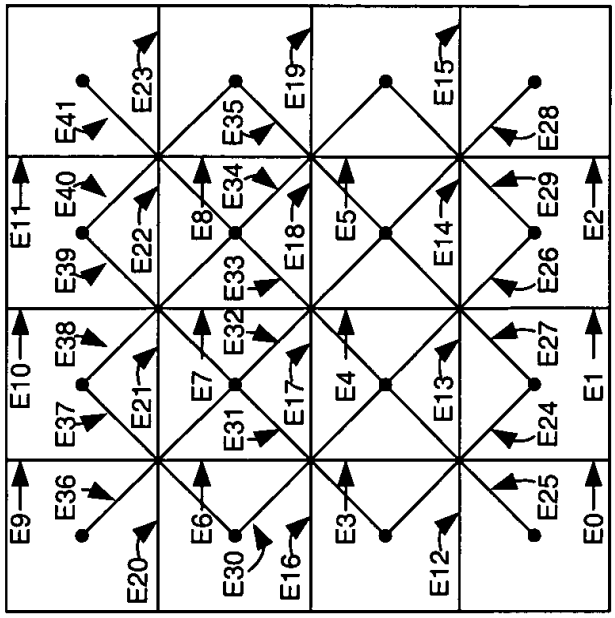

Figure 10

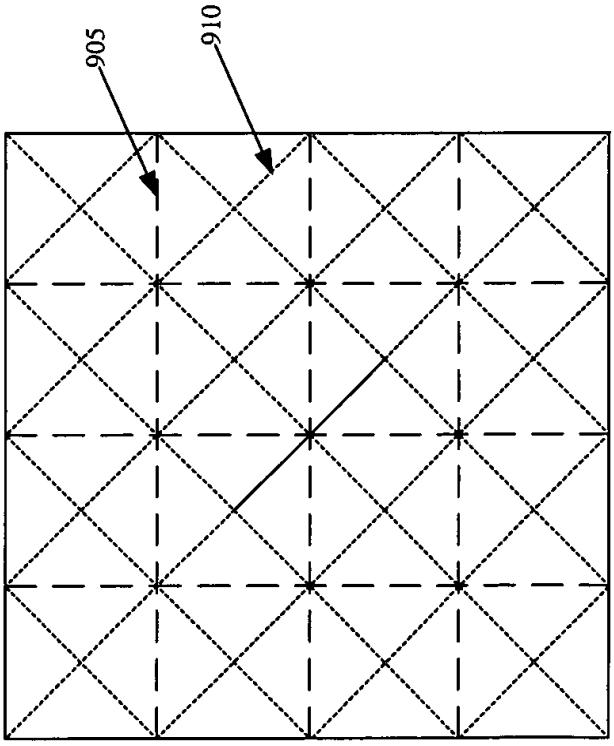

Figure 9

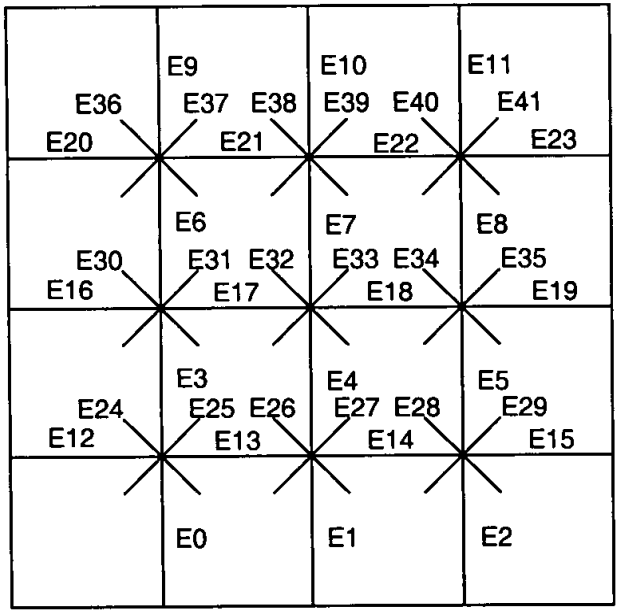

Figure 11

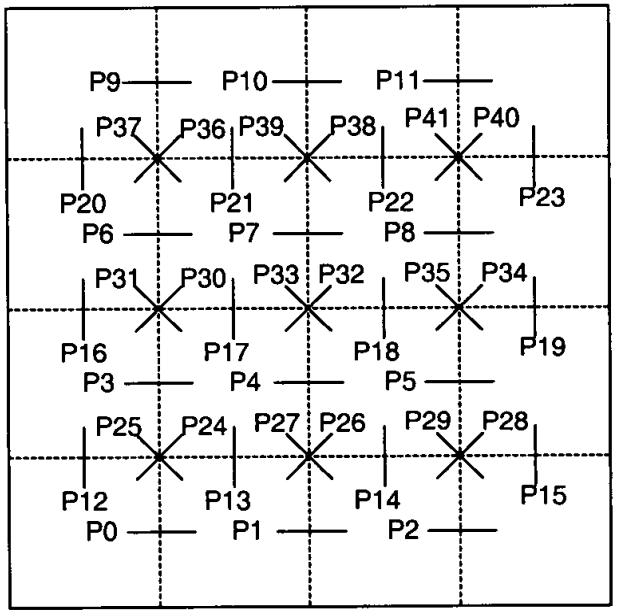

Figure 12

10025618.041702

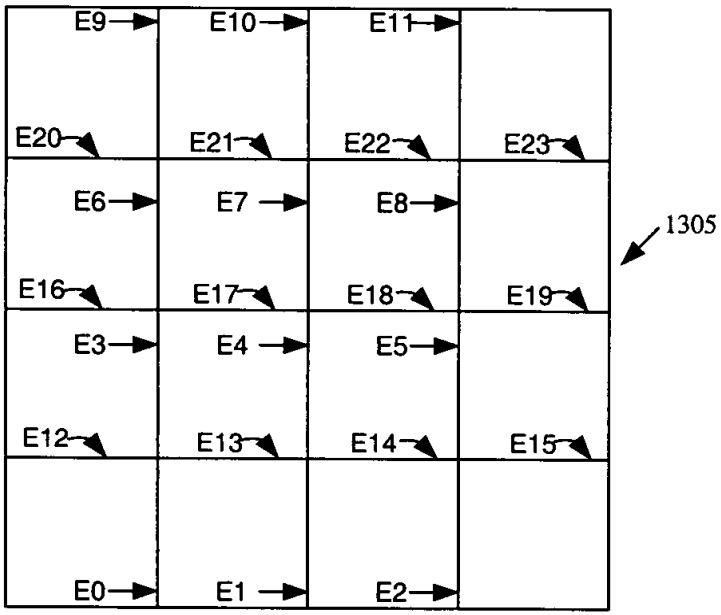

Figure 13

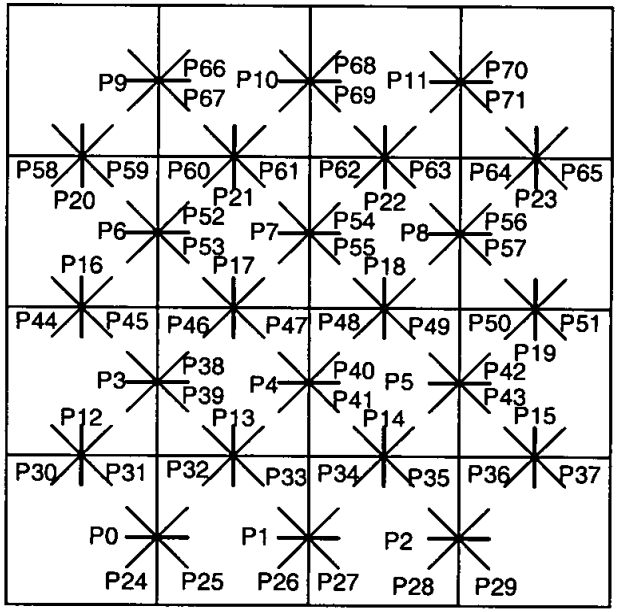

Figure 14

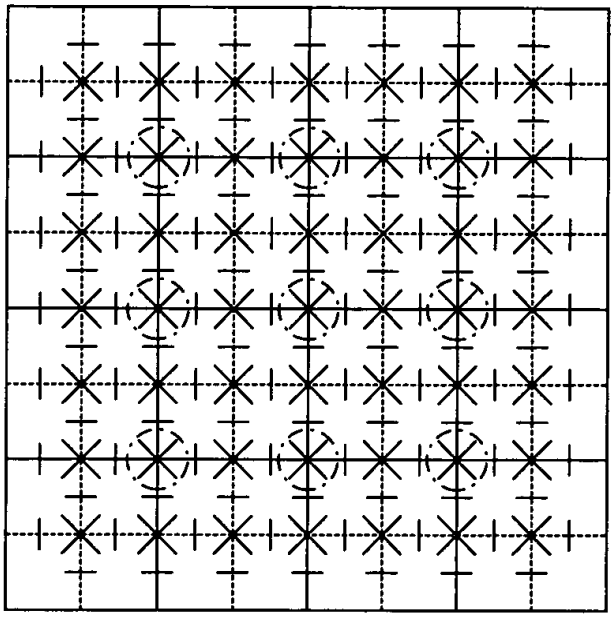

Figure 17

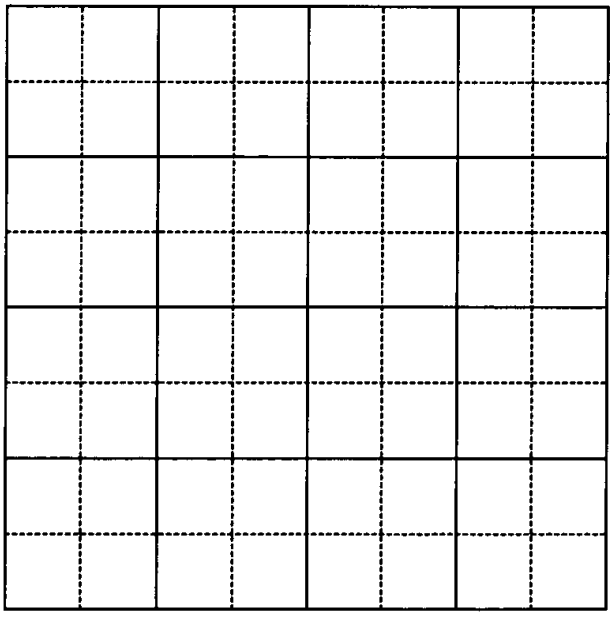

Figure 15

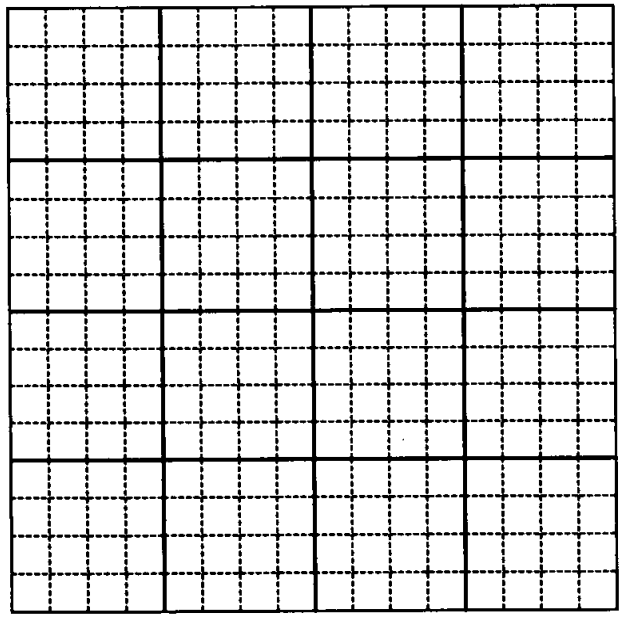

Figure 16

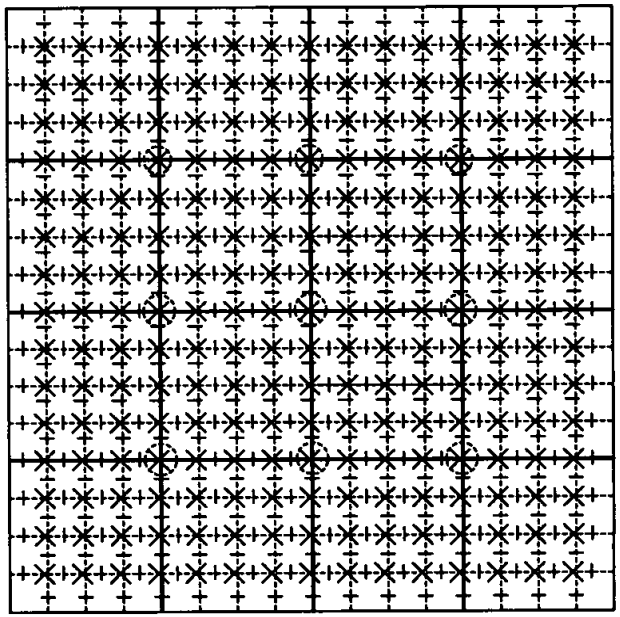

Figure 18

10025618-041702

10039618.041702

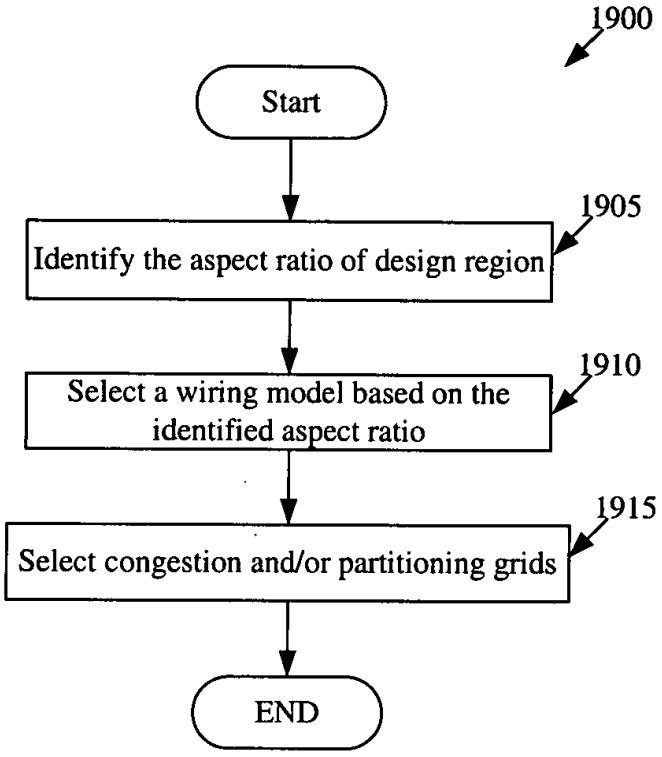

Figure 19

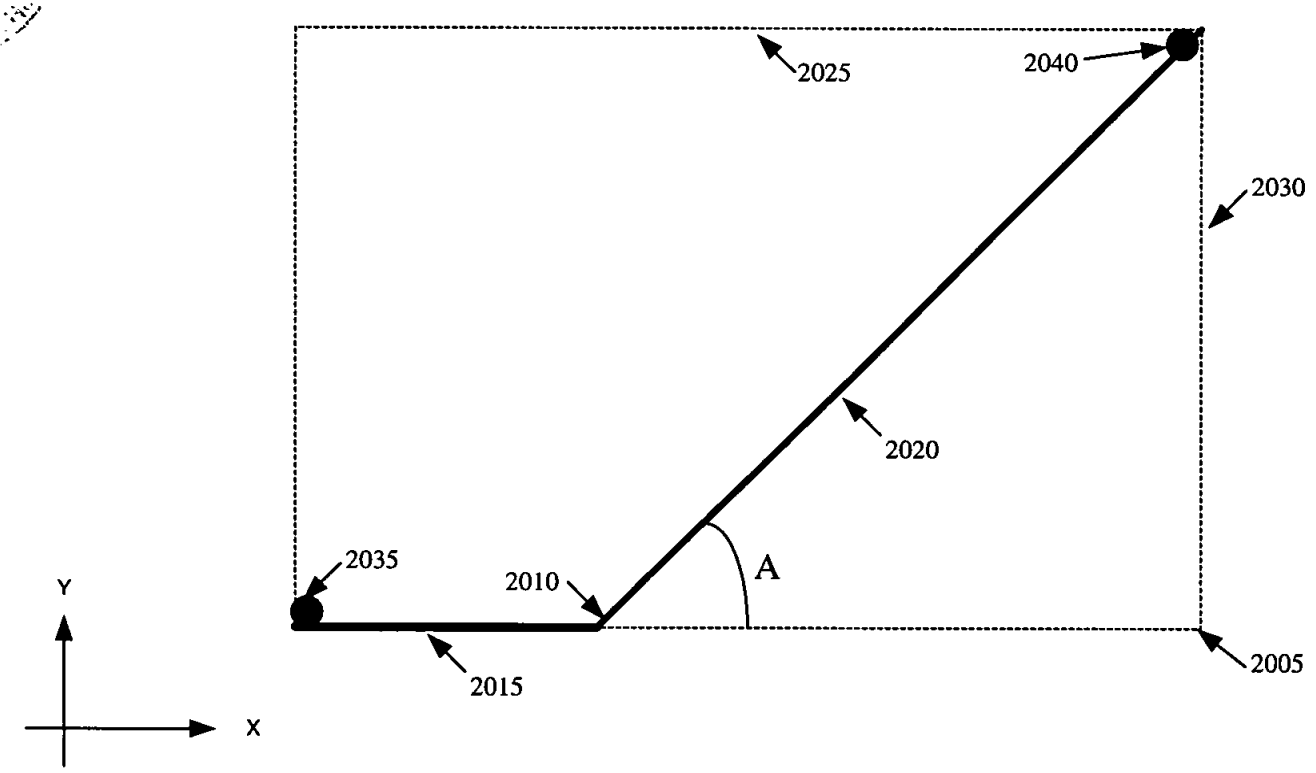

Figure 20

1002961.0141702

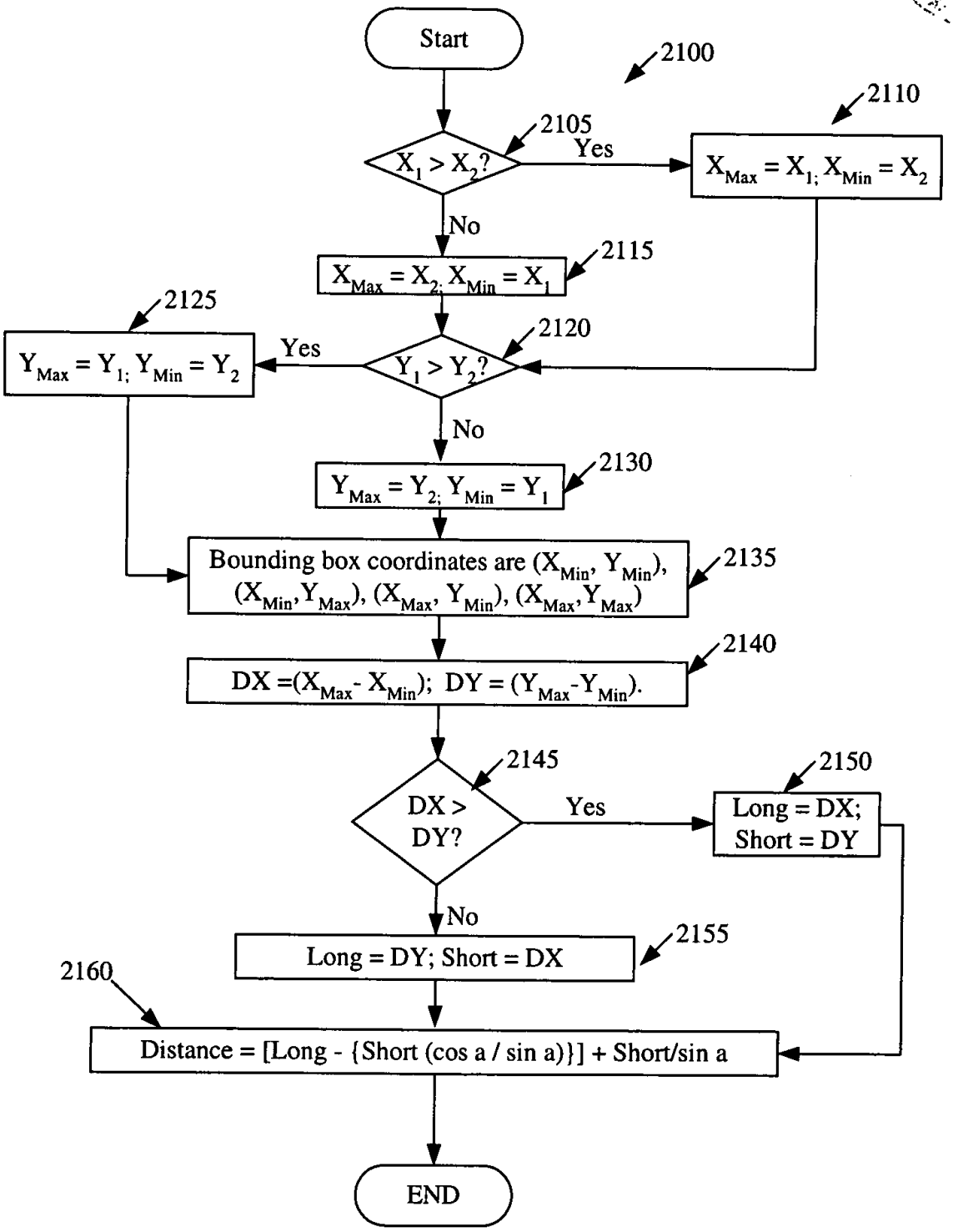
Start
2100
2105
$X_1 > X_2$?
Yes
2110
$X_{Max} = X_1$; $X_{Min} = X_2$
No
2115
$X_{Max} = X_2$; $X_{Min} = X_1$
2125
$Y_{Max} = Y_1$; $Y_{Min} = Y_2$
Yes
2120
$Y_1 > Y_2$?
No
2130
$Y_{Max} = Y_2$; $Y_{Min} = Y_1$
Bounding box coordinates are (X_{Min}, Y_{Min}), (X_{Min},Y_{Max}), (X_{Max}, Y_{Min}), (X_{Max},Y_{Max})
2135
DX =(X_{Max}- X_{Min}); DY = (Y_{Max}-Y_{Min}).
2140
2145
DX > DY?
Yes
2150
Long = DX; Short = DY
No
2155
Long = DY; Short = DX
2160
Distance = [Long - {Short (cos a / sin a)}] + Short/sin a
END

Figure 21

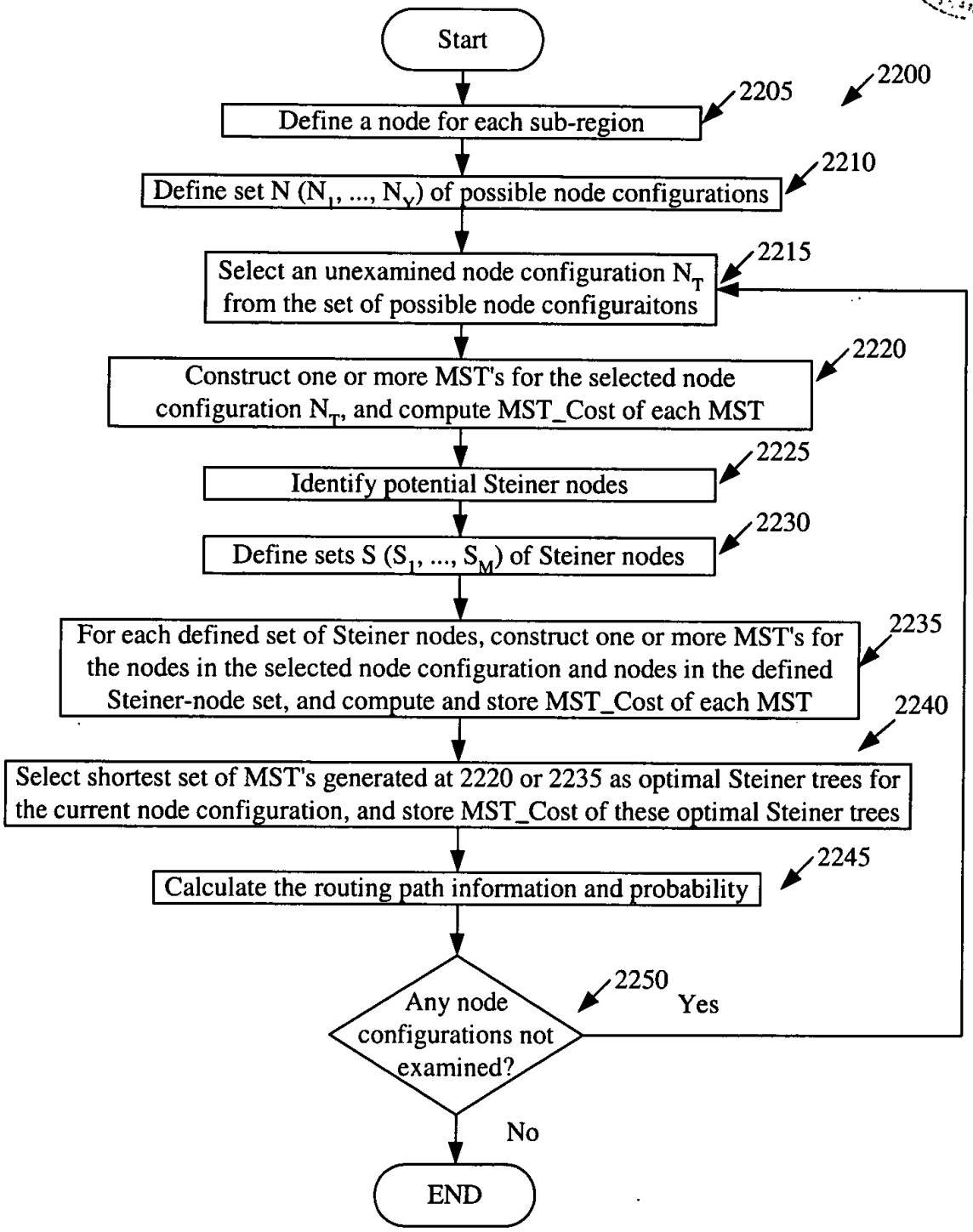
Start
2200
2205
Define a node for each sub-region
2210
Define set N (N_1, ..., N_Y) of possible node configurations
2215
Select an unexamined node configuration N_T from the set of possible node configuraitons
2220
Construct one or more MST's for the selected node configuration N_T, and compute MST_Cost of each MST
2225
Identify potential Steiner nodes
2230
Define sets S (S_1, ..., S_M) of Steiner nodes
2235
For each defined set of Steiner nodes, construct one or more MST's for the nodes in the selected node configuration and nodes in the defined Steiner-node set, and compute and store MST_Cost of each MST
2240
Select shortest set of MST's generated at 2220 or 2235 as optimal Steiner trees for the current node configuration, and store MST_Cost of these optimal Steiner trees
2245
Calculate the routing path information and probability
2250
Any node configurations not examined?
Yes
No
END

Figure 22

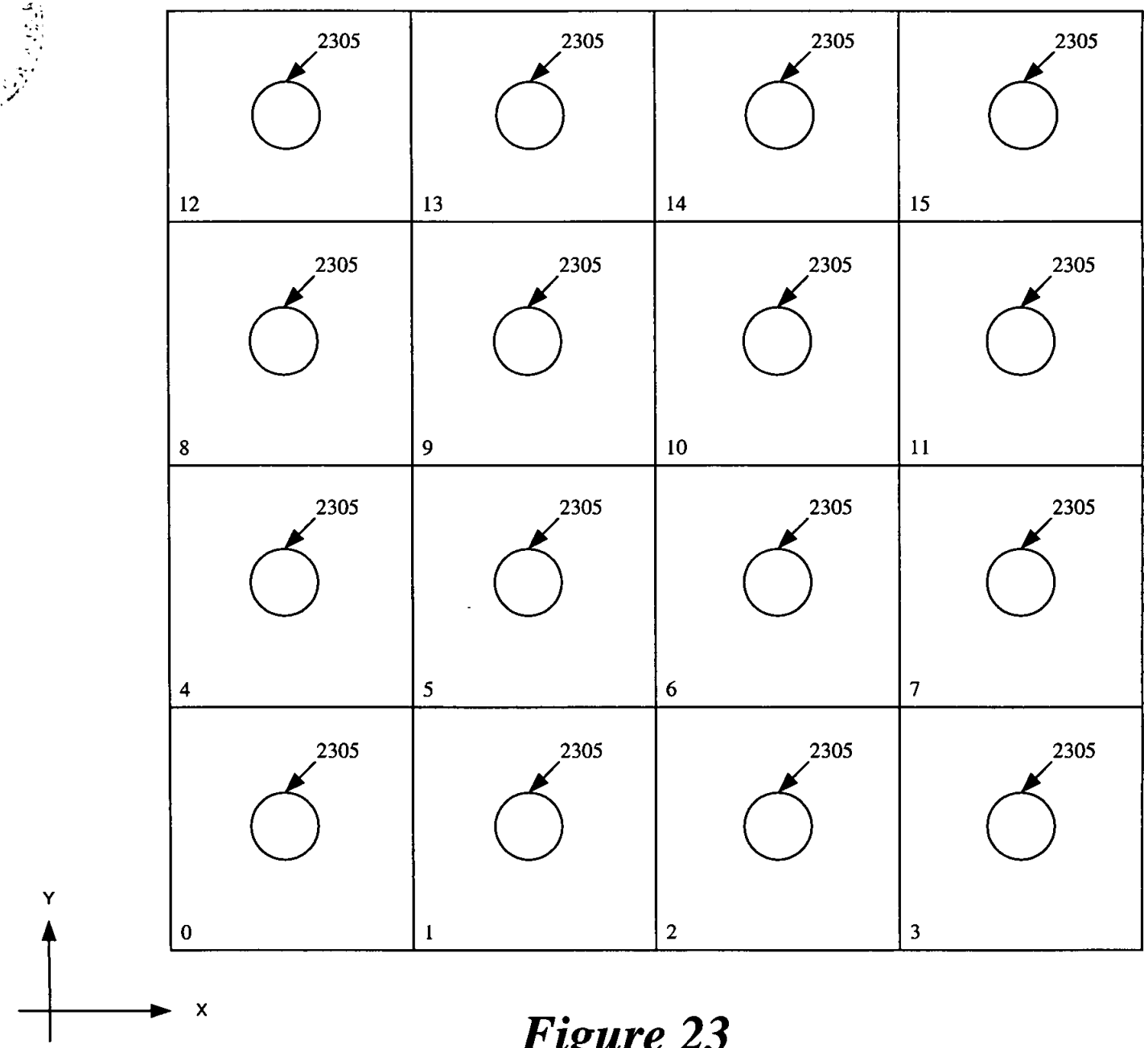

Figure 23

10039618.041702

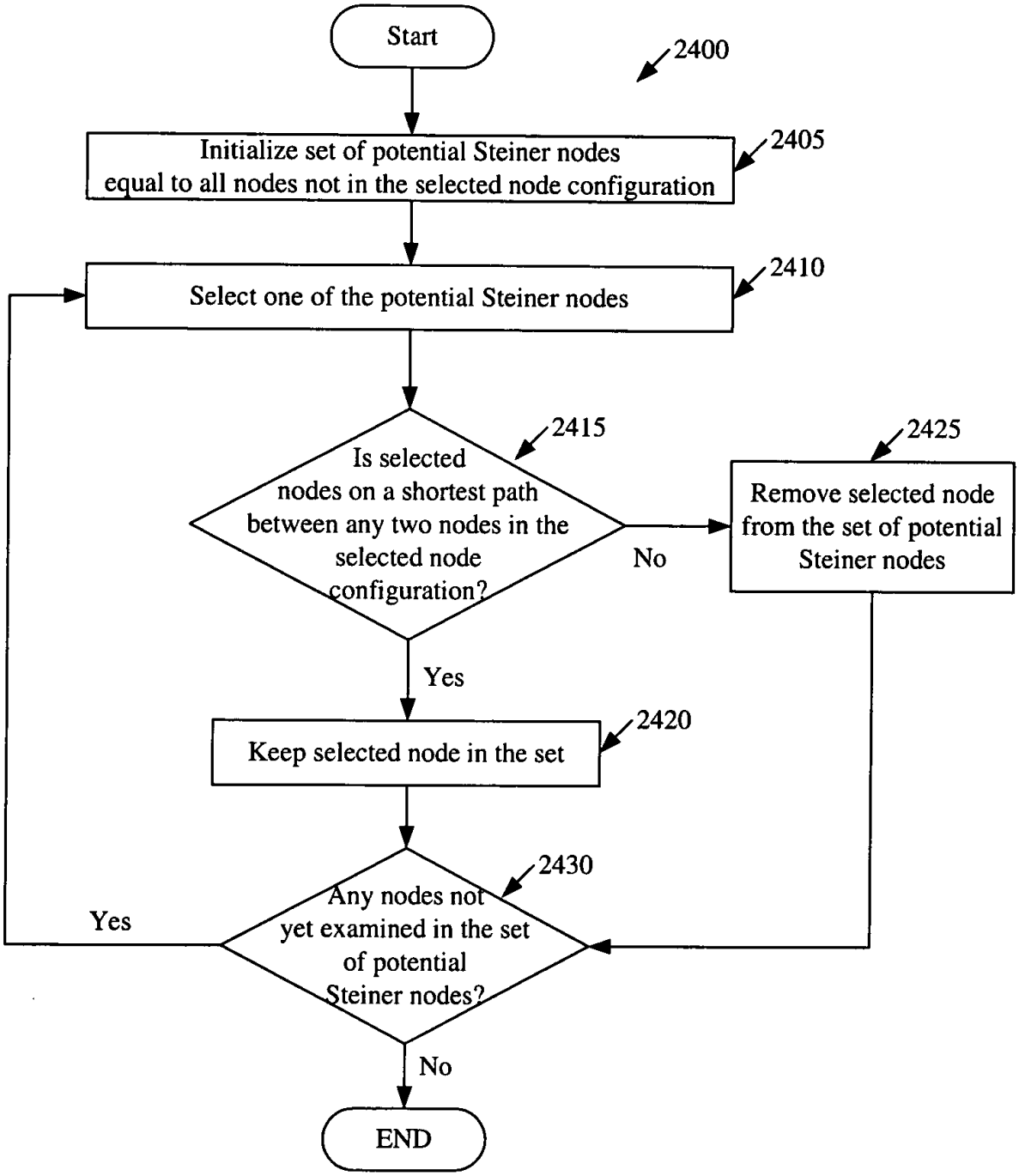

Figure 24

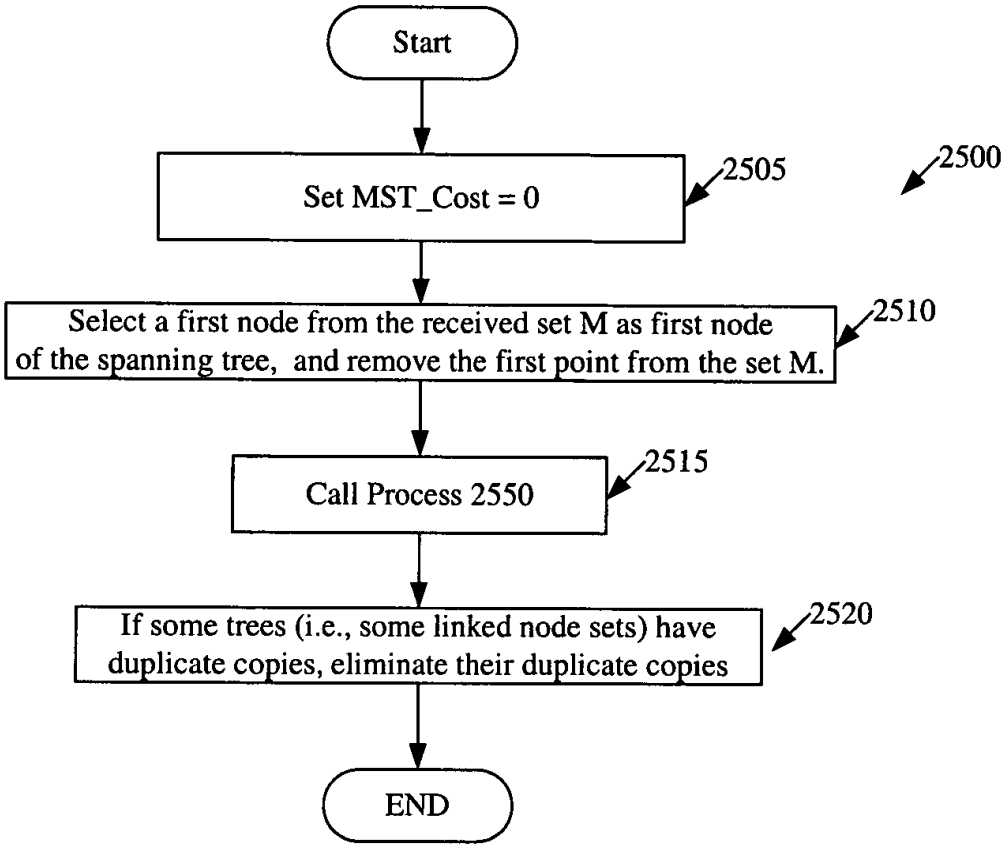
Start
Set MST_Cost = 0
2505
2500
Select a first node from the received set M as first node of the spanning tree, and remove the first point from the set M.
2510
Call Process 2550
2515
If some trees (i.e., some linked node sets) have duplicate copies, eliminate their duplicate copies
2520
END

Figure 25A

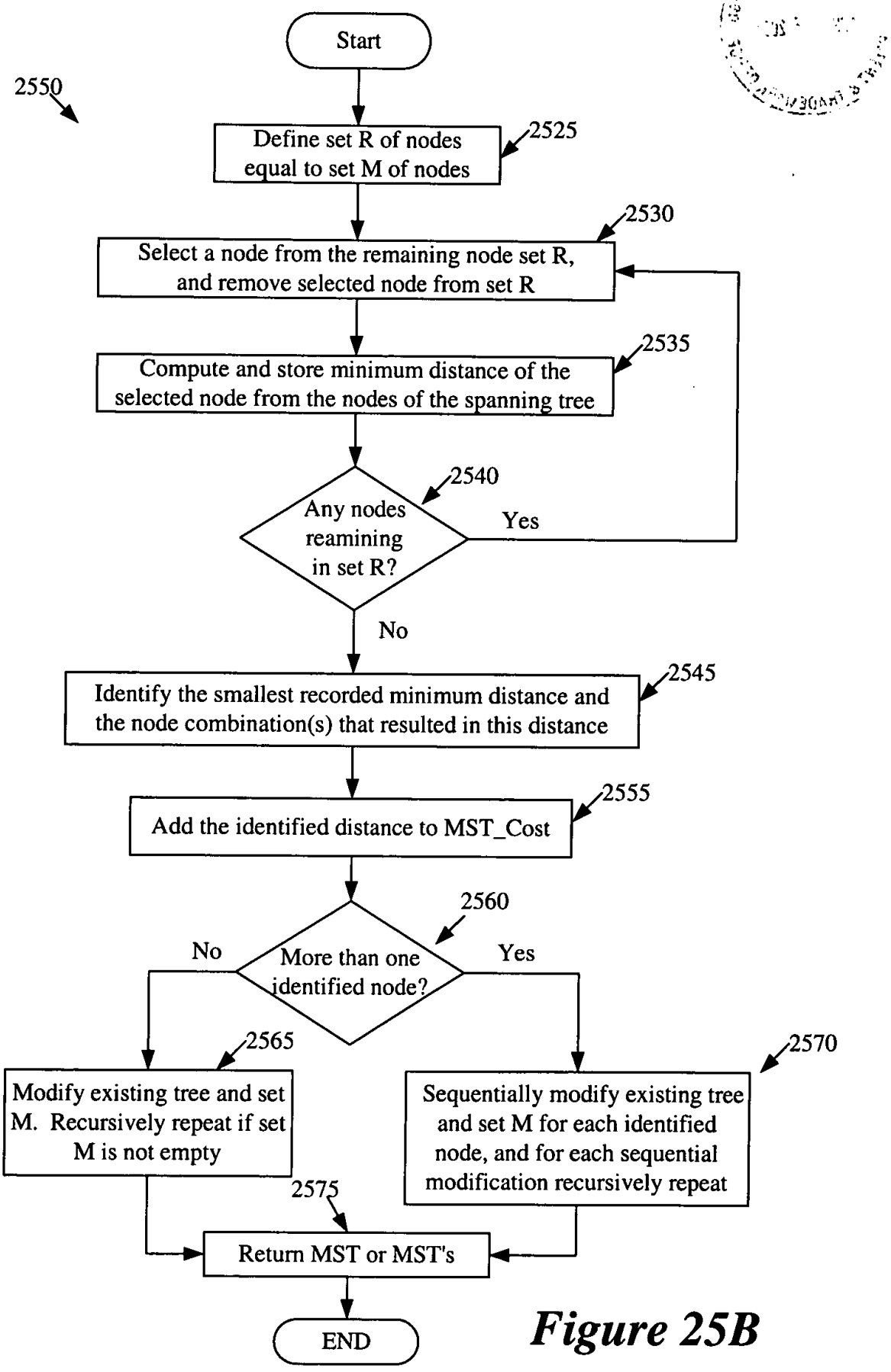
Start
2550
Define set R of nodes equal to set M of nodes
2525
2530
Select a node from the remaining node set R, and remove selected node from set R
2535
Compute and store minimum distance of the selected node from the nodes of the spanning tree
2540
Any nodes reamining in set R?
Yes
No
2545
Identify the smallest recorded minimum distance and the node combination(s) that resulted in this distance
2555
Add the identified distance to MST_Cost
2560
More than one identified node?
No
Yes
2565
Modify existing tree and set M. Recursively repeat if set M is not empty
2570
Sequentially modify existing tree and set M for each identified node, and for each sequential modification recursively repeat
2575
Return MST or MST's
END

Figure 25B

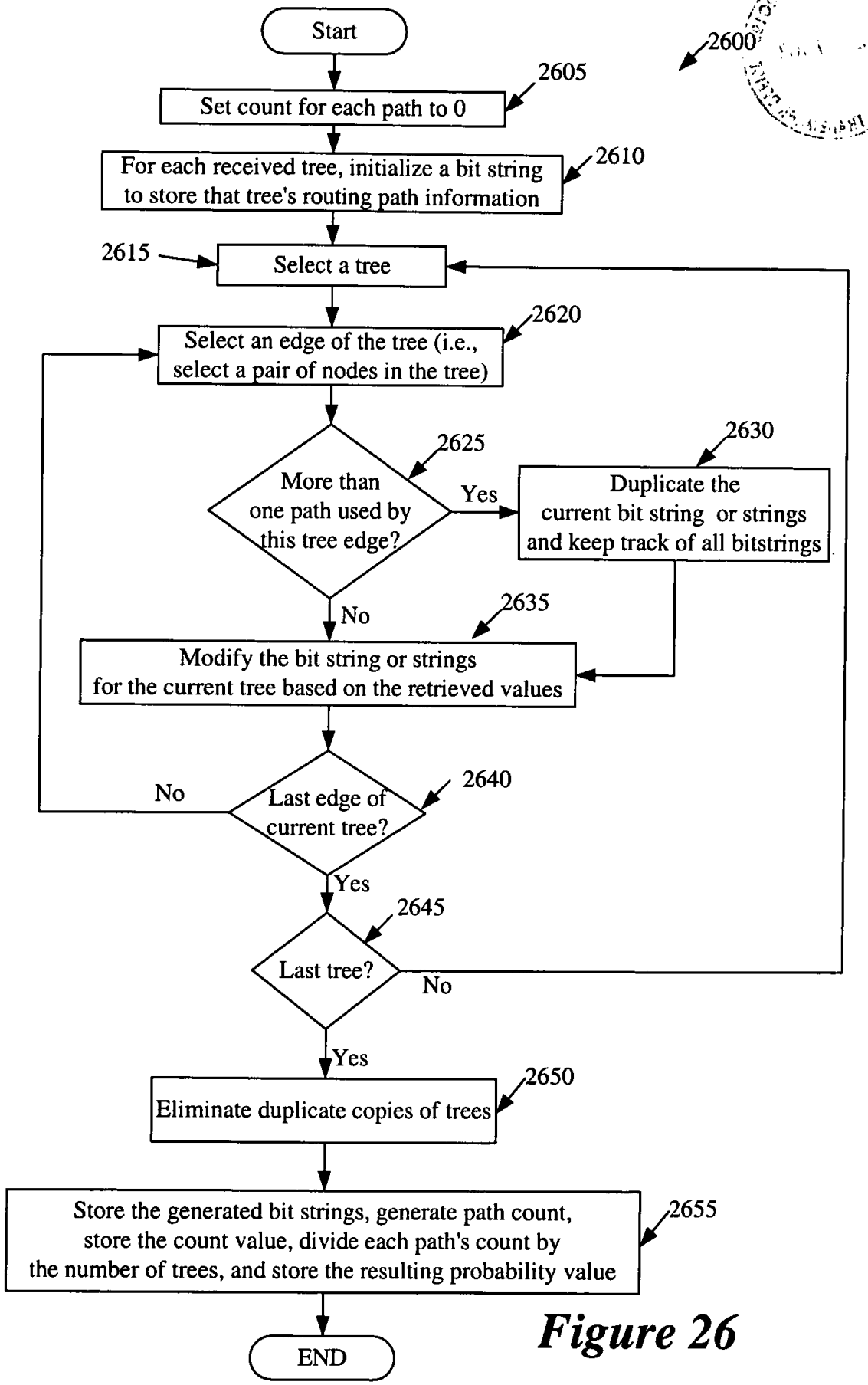
Start
2600
2605
Set count for each path to 0
2610
For each received tree, initialize a bit string to store that tree's routing path information
2615
Select a tree
2620
Select an edge of the tree (i.e., select a pair of nodes in the tree)
2625
More than one path used by this tree edge?
Yes
2630
Duplicate the current bit string or strings and keep track of all bitstrings
No
2635
Modify the bit string or strings for the current tree based on the retrieved values
2640
Last edge of current tree?
No
Yes
2645
Last tree?
No
Yes
2650
Eliminate duplicate copies of trees
2655
Store the generated bit strings, generate path count, store the count value, divide each path's count by the number of trees, and store the resulting probability value
END

Figure 26

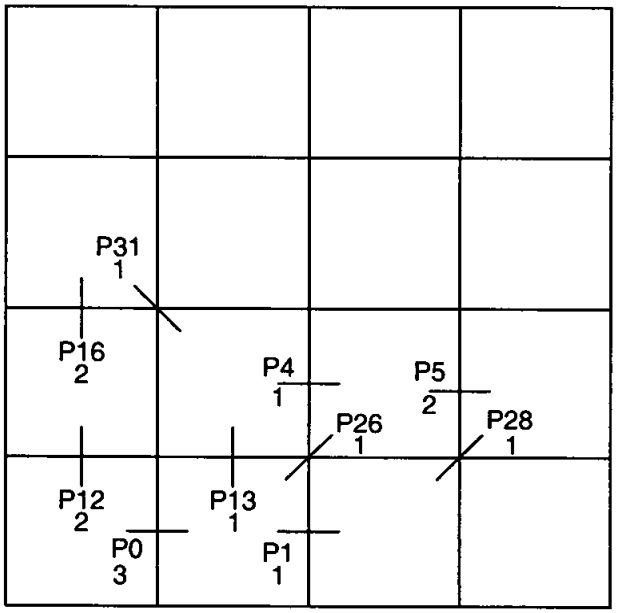

Figure 27

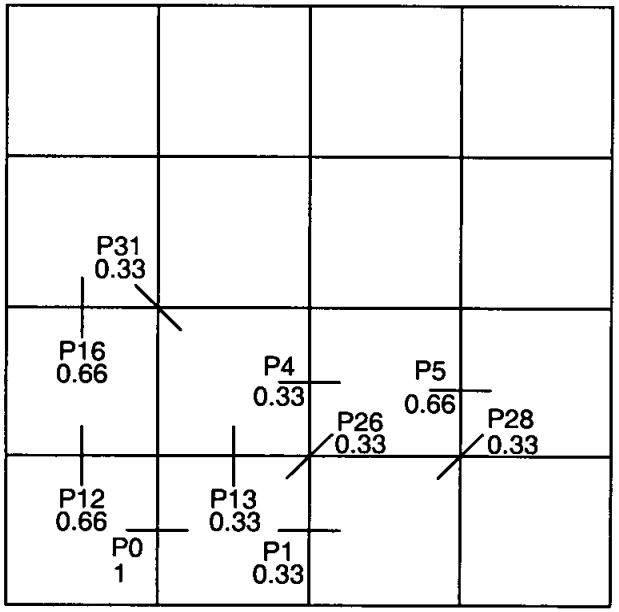

Figure 28

20/11/08 08:56:00

10039618.041702

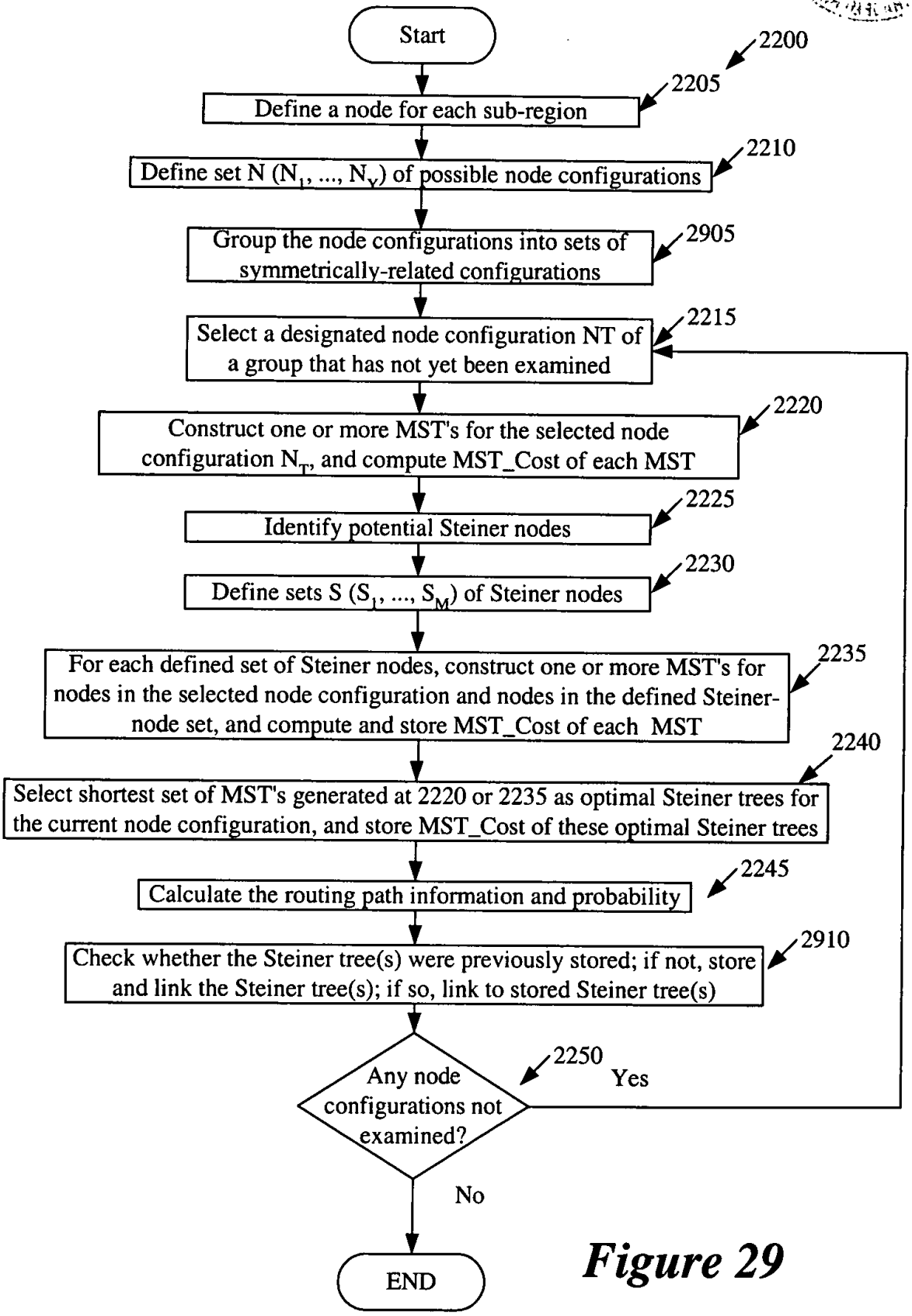

Figure 29

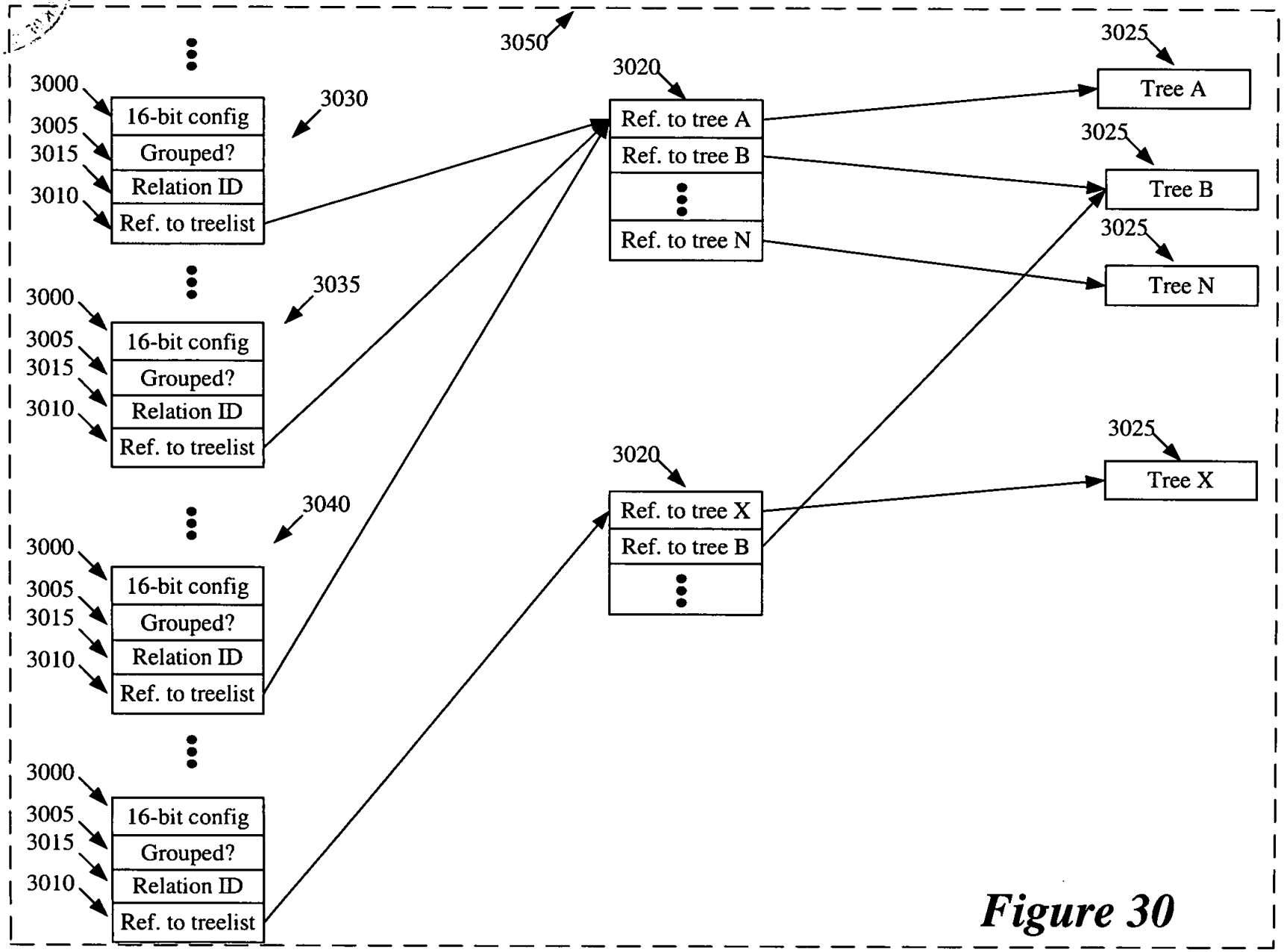

Figure 30

10025618 - 041702

10039618 .041702

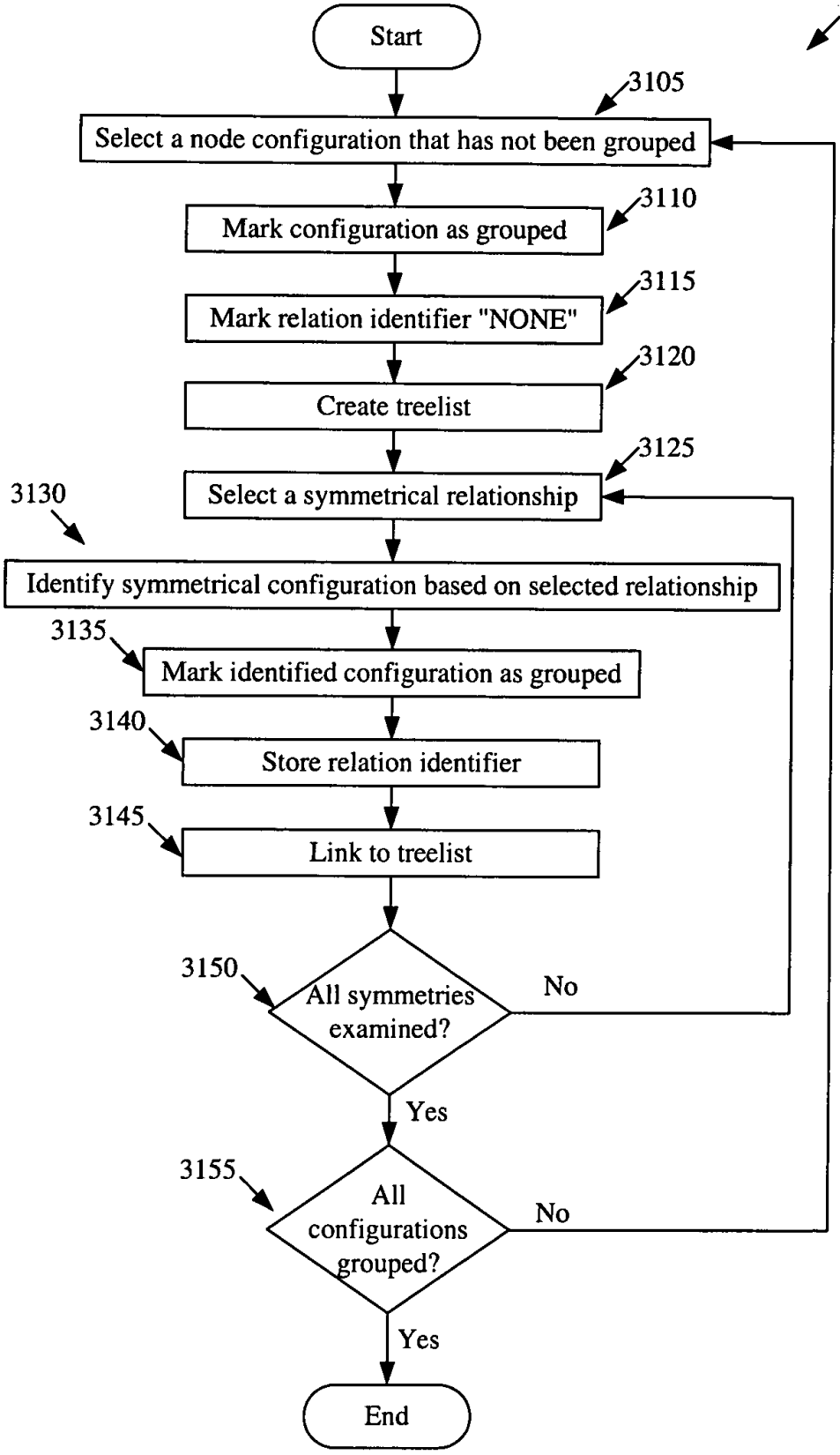

Figure 31

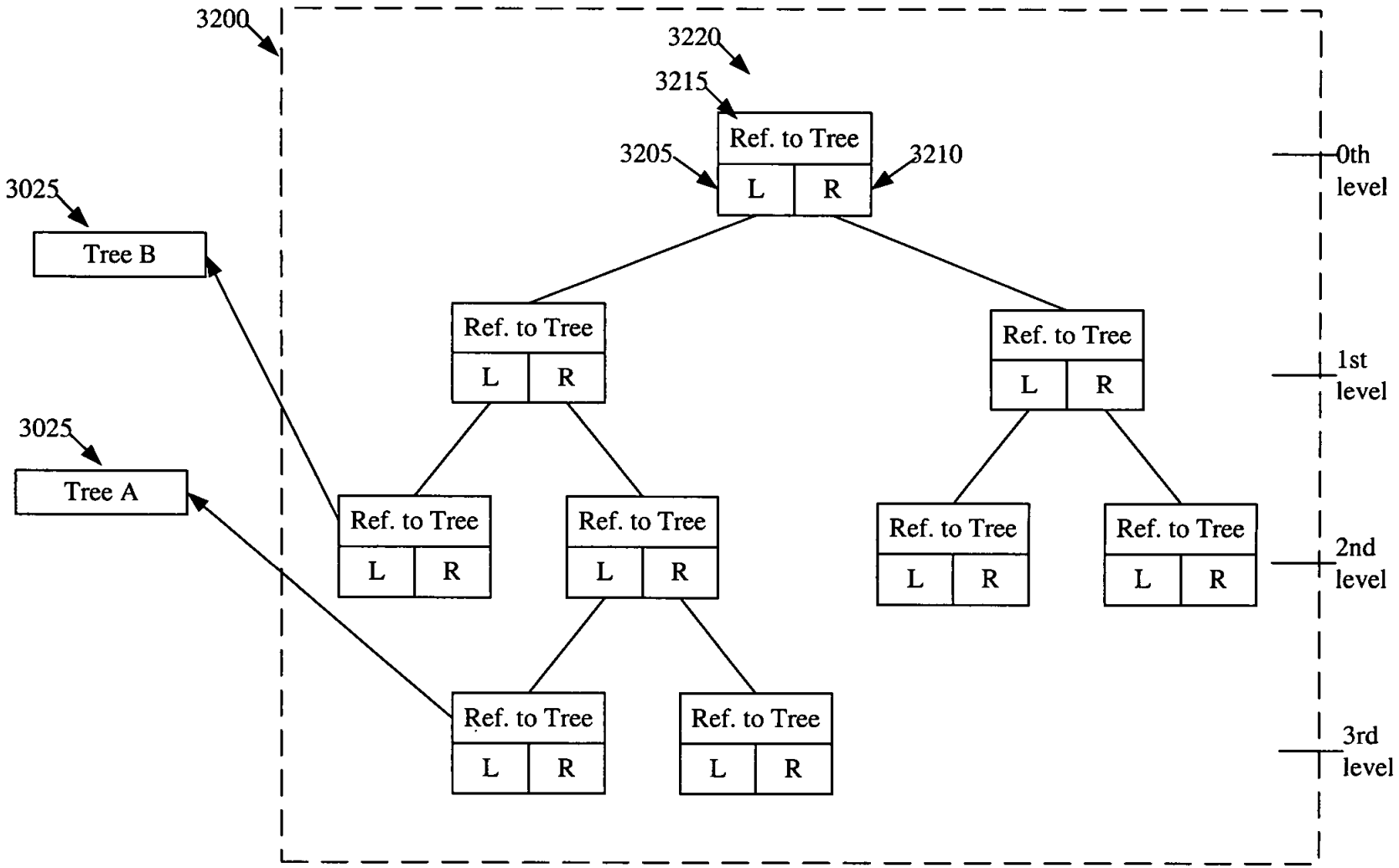

Figure 32

10035618.041702

10039618.041702

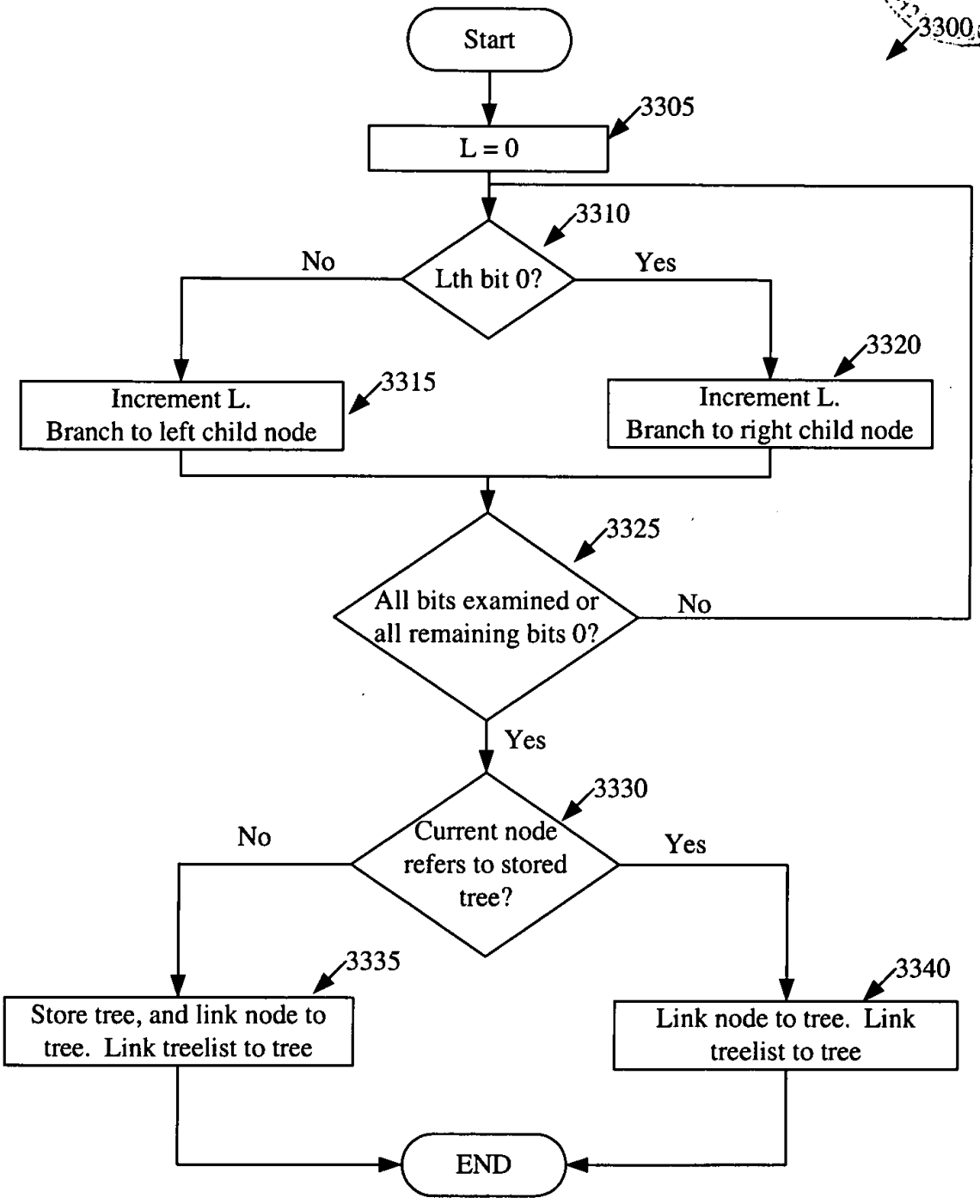

Figure 33

10039618.041702

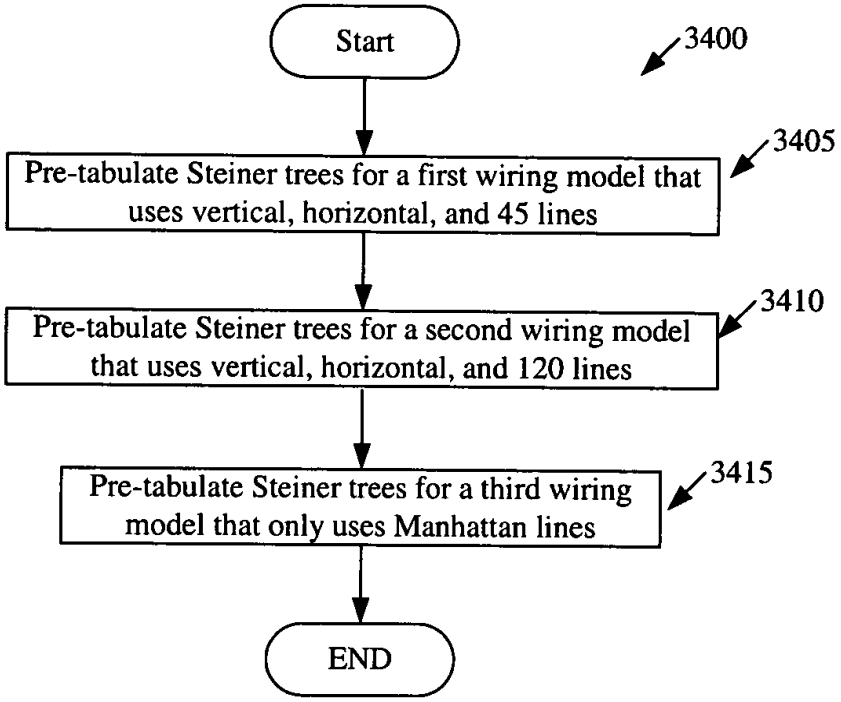

Figure 34

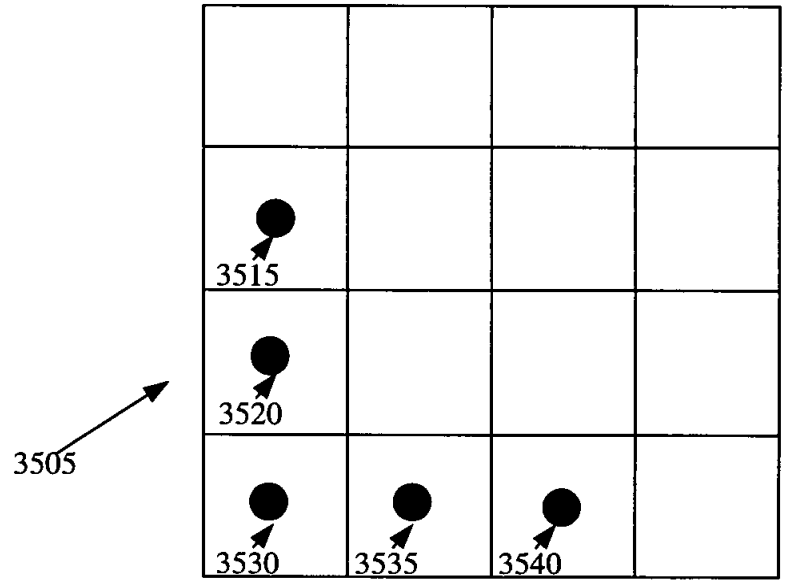

Figure 35A

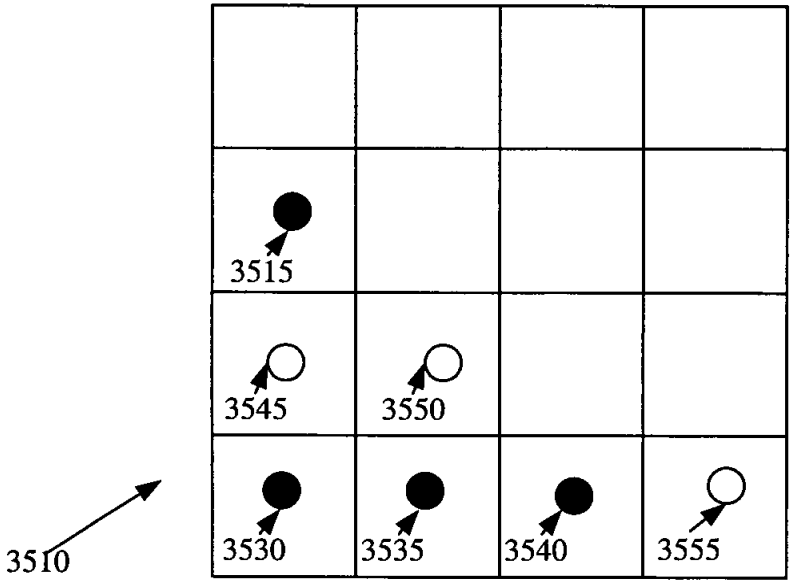

Figure 35B

1002618.04702

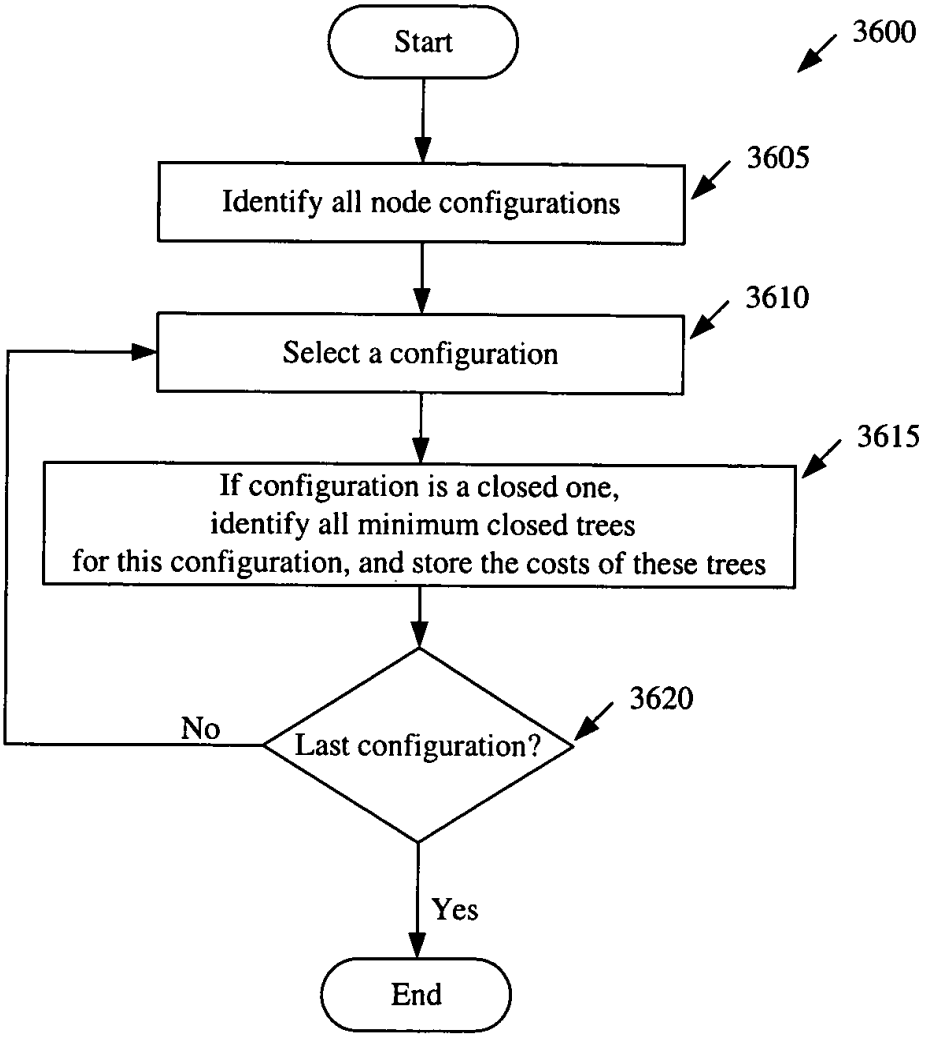
Start
3600
Identify all node configurations
3605
Select a configuration
3610
If configuration is a closed one,
identify all minimum closed trees
for this configuration, and store the costs of these trees
3615
Last configuration?
3620
No
Yes
End

Figure 36

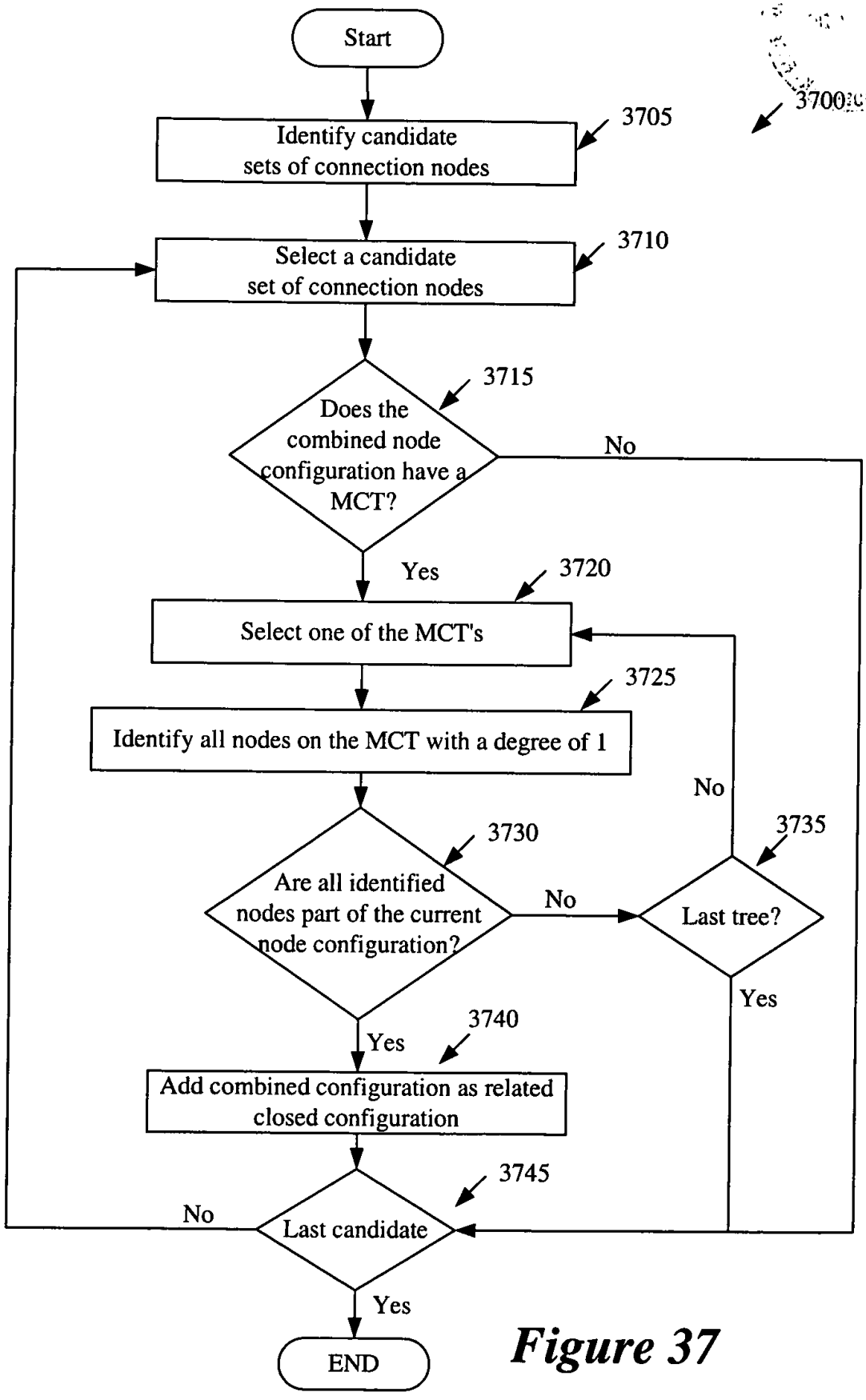
Start
Identify candidate sets of connection nodes
3705
Select a candidate set of connection nodes
3710
Does the combined node configuration have a MCT?
3715
No
Yes
Select one of the MCT's
3720
Identify all nodes on the MCT with a degree of 1
3725
Are all identified nodes part of the current node configuration?
3730
No
Last tree?
3735
No
Yes
Yes
Add combined configuration as related closed configuration
3740
Last candidate
3745
No
Yes
END

Figure 37

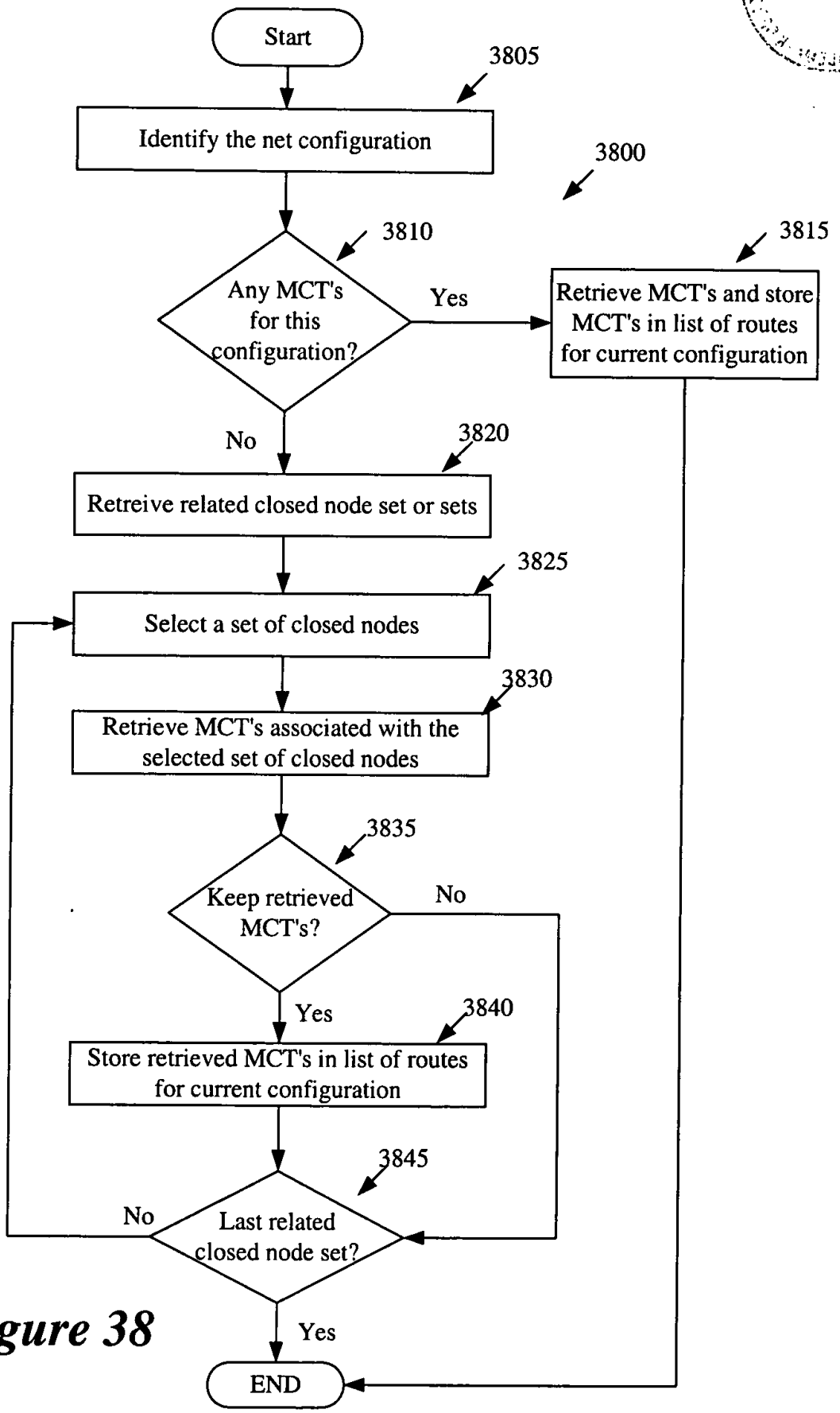
Start
3805
Identify the net configuration
3800
3810
Any MCT's for this configuration?
Yes
3815
Retrieve MCT's and store MCT's in list of routes for current configuration
No
3820
Retreive related closed node set or sets
3825
Select a set of closed nodes
3830
Retrieve MCT's associated with the selected set of closed nodes
3835
Keep retrieved MCT's?
No
Yes
3840
Store retrieved MCT's in list of routes for current configuration
3845
No
Last related closed node set?
Yes
END

Figure 38

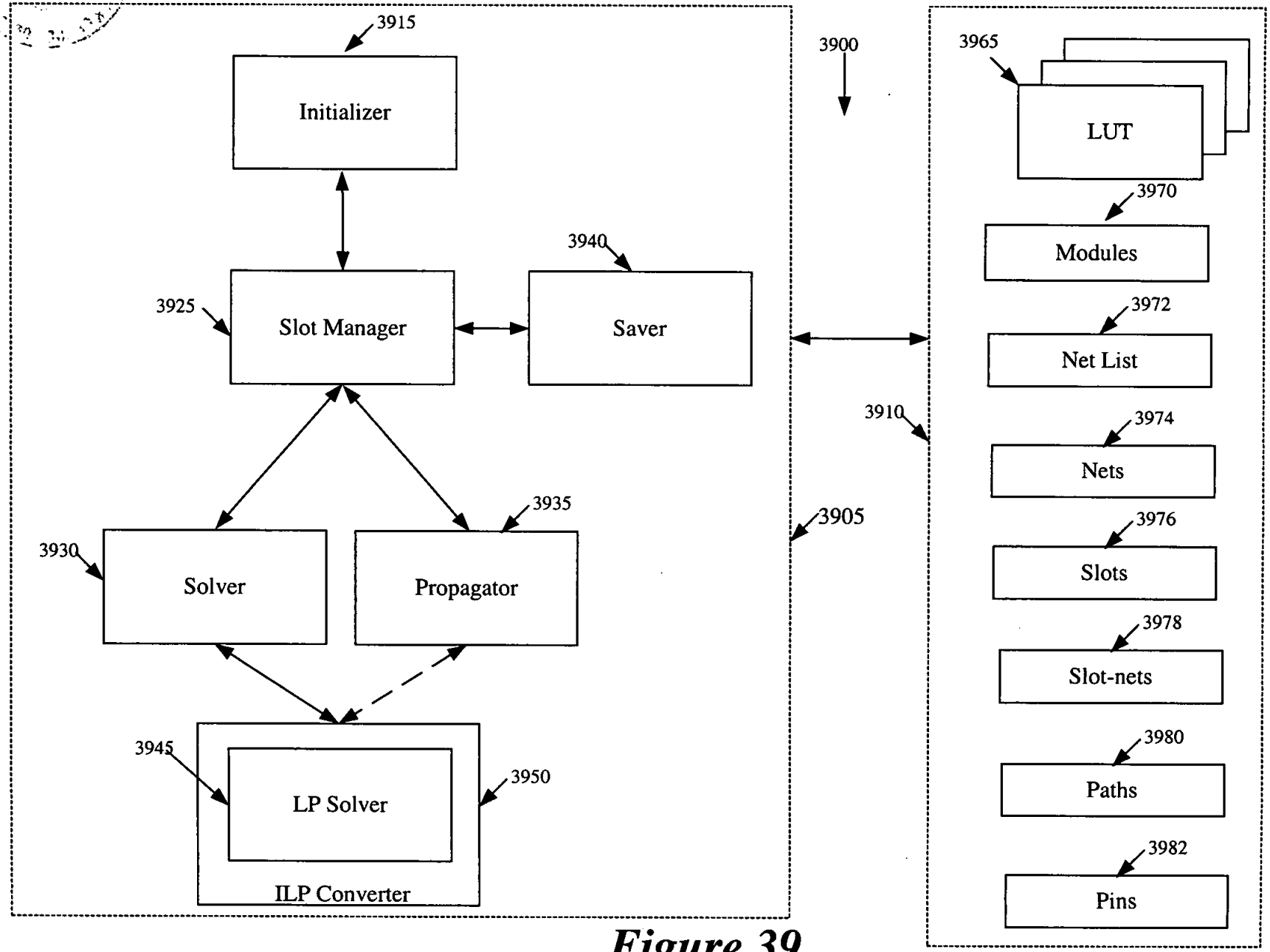
3915
Initializer
3900
3965
LUT
3970
Modules
3940
3925
Slot Manager
Saver
3972
Net List
3910
3974
Nets
3935
3905
3976
Slots
3930
Solver
Propagator
3978
Slot-nets
3945
LP Solver
3950
3980
Paths
ILP Converter
3982
Pins

Figure 39

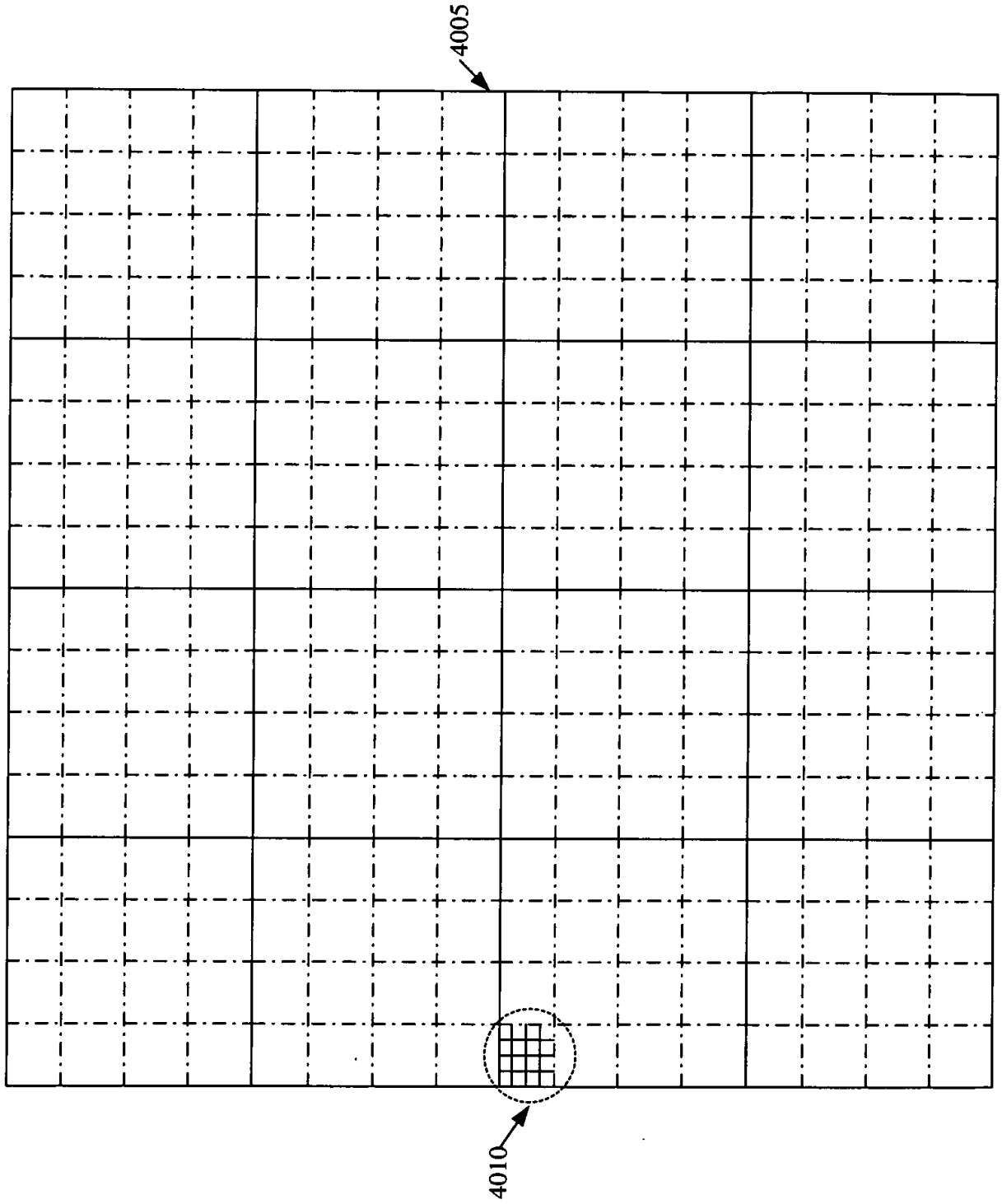

Figure 40

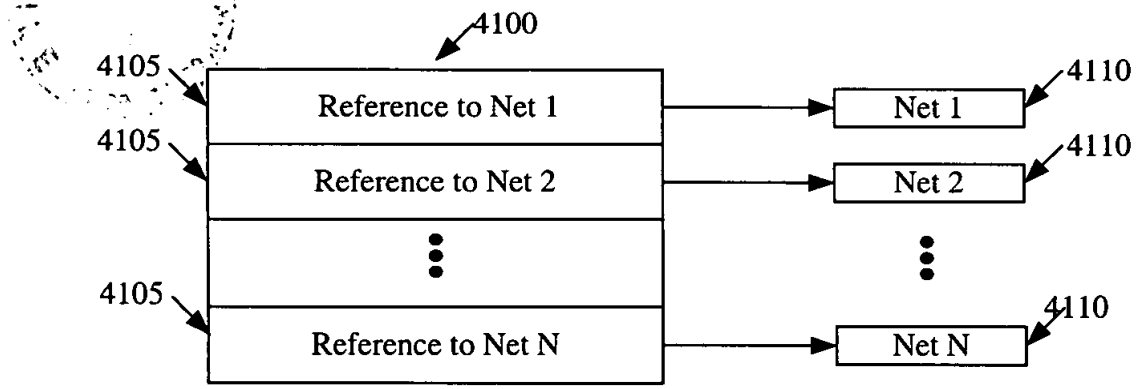

Figure 41

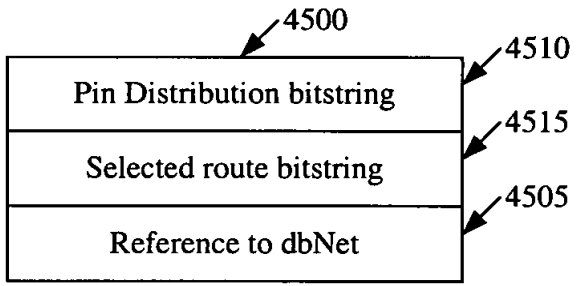

Figure 45

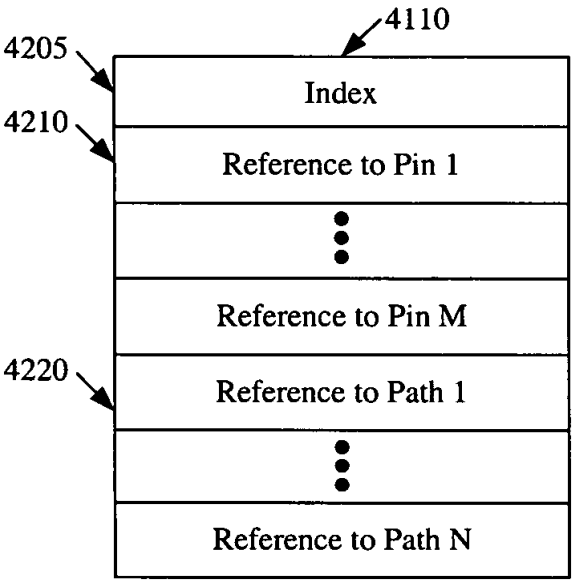

Figure 42

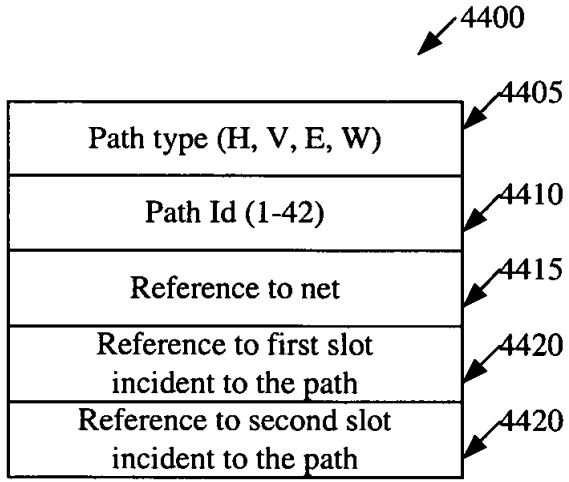

Figure 44

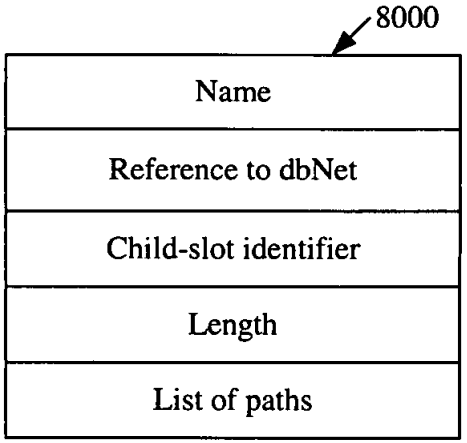

Figure 80

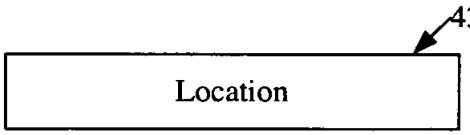

Figure 43

10025618.041702

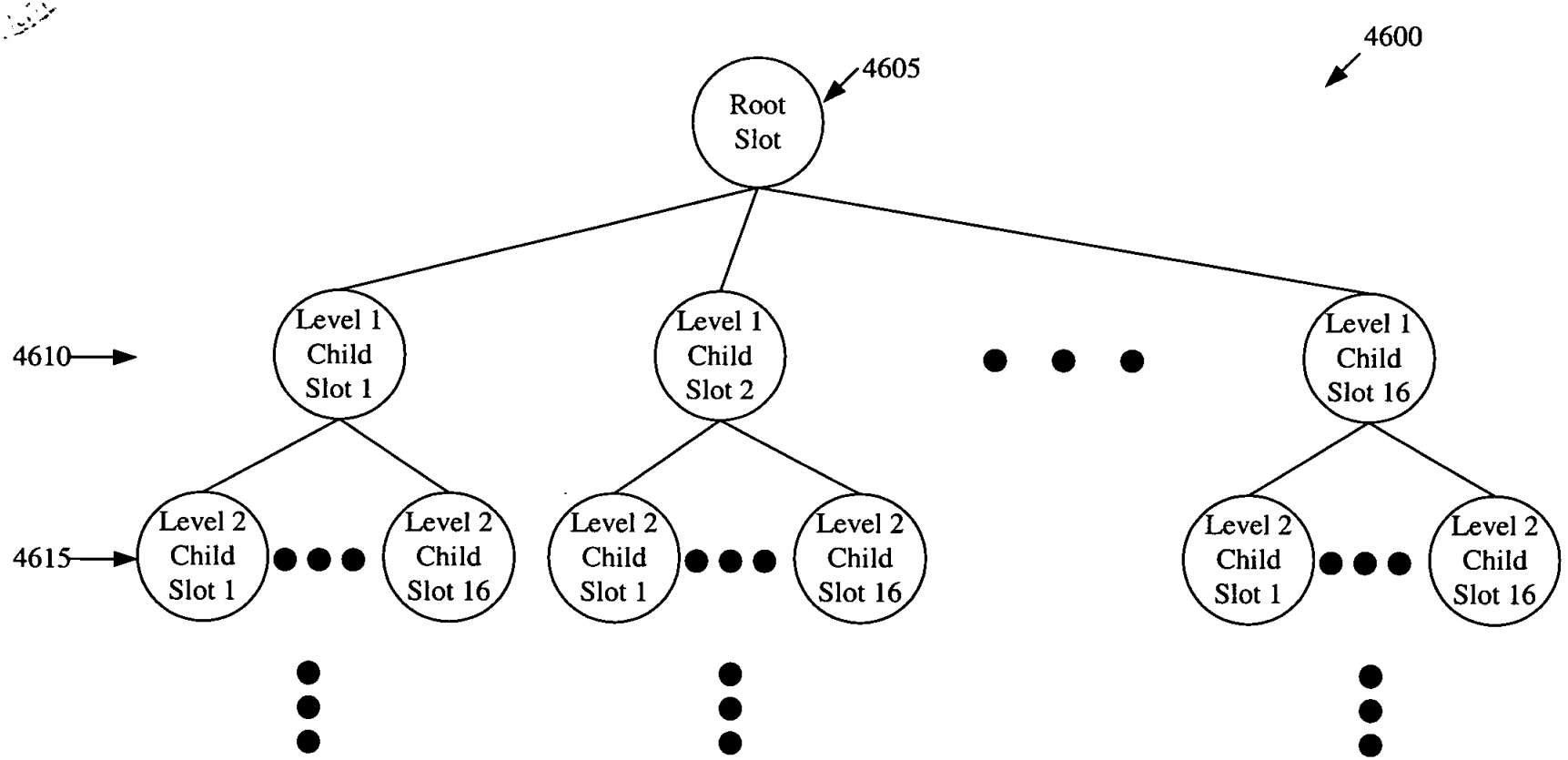

Figure 46

10020518-041702

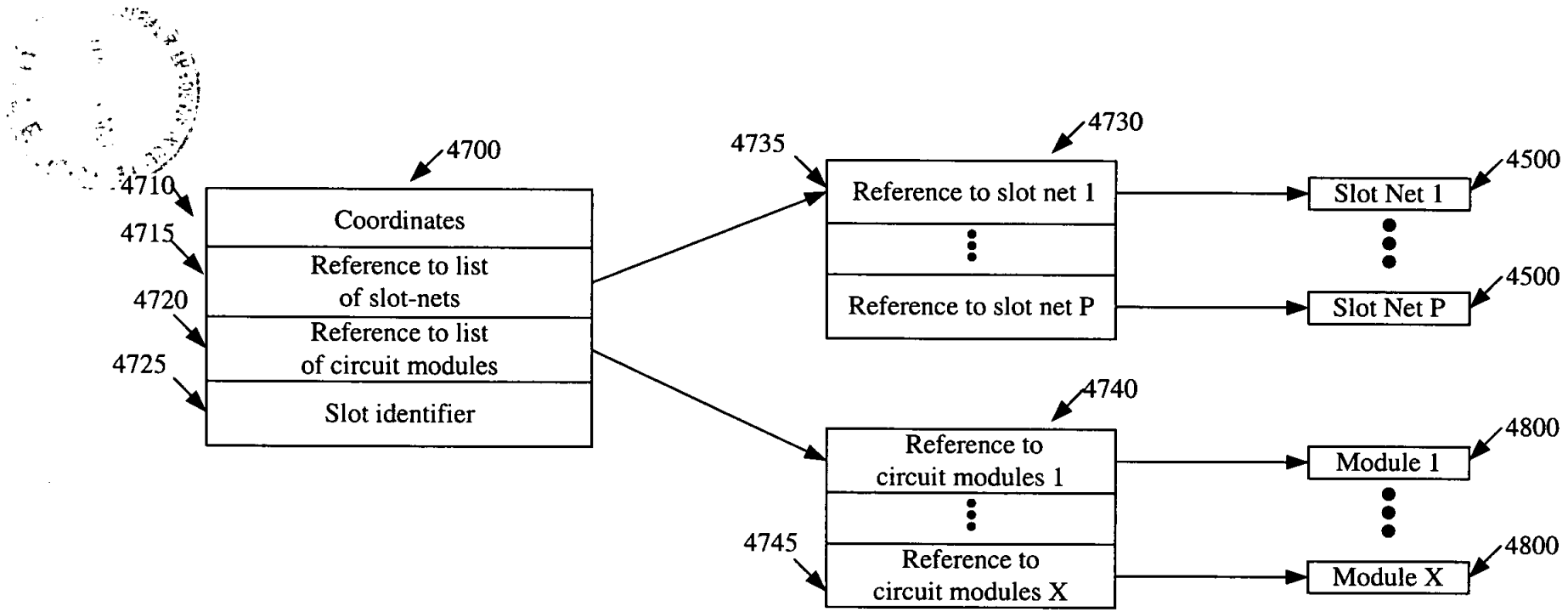

Figure 47

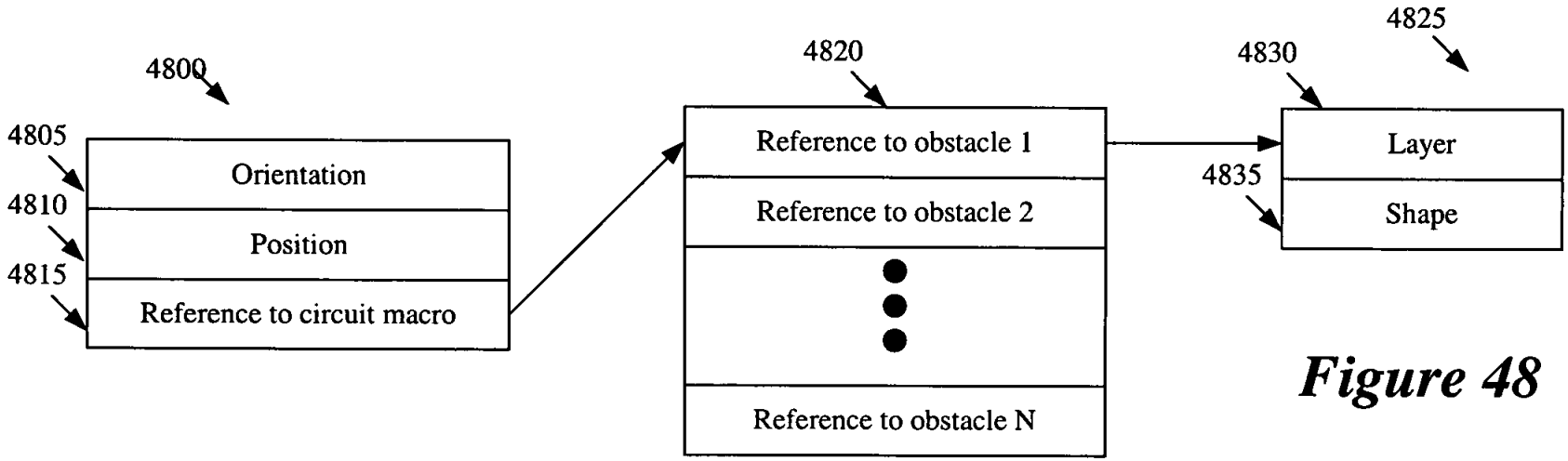

Figure 48

20070048-041702

10079618 .041702

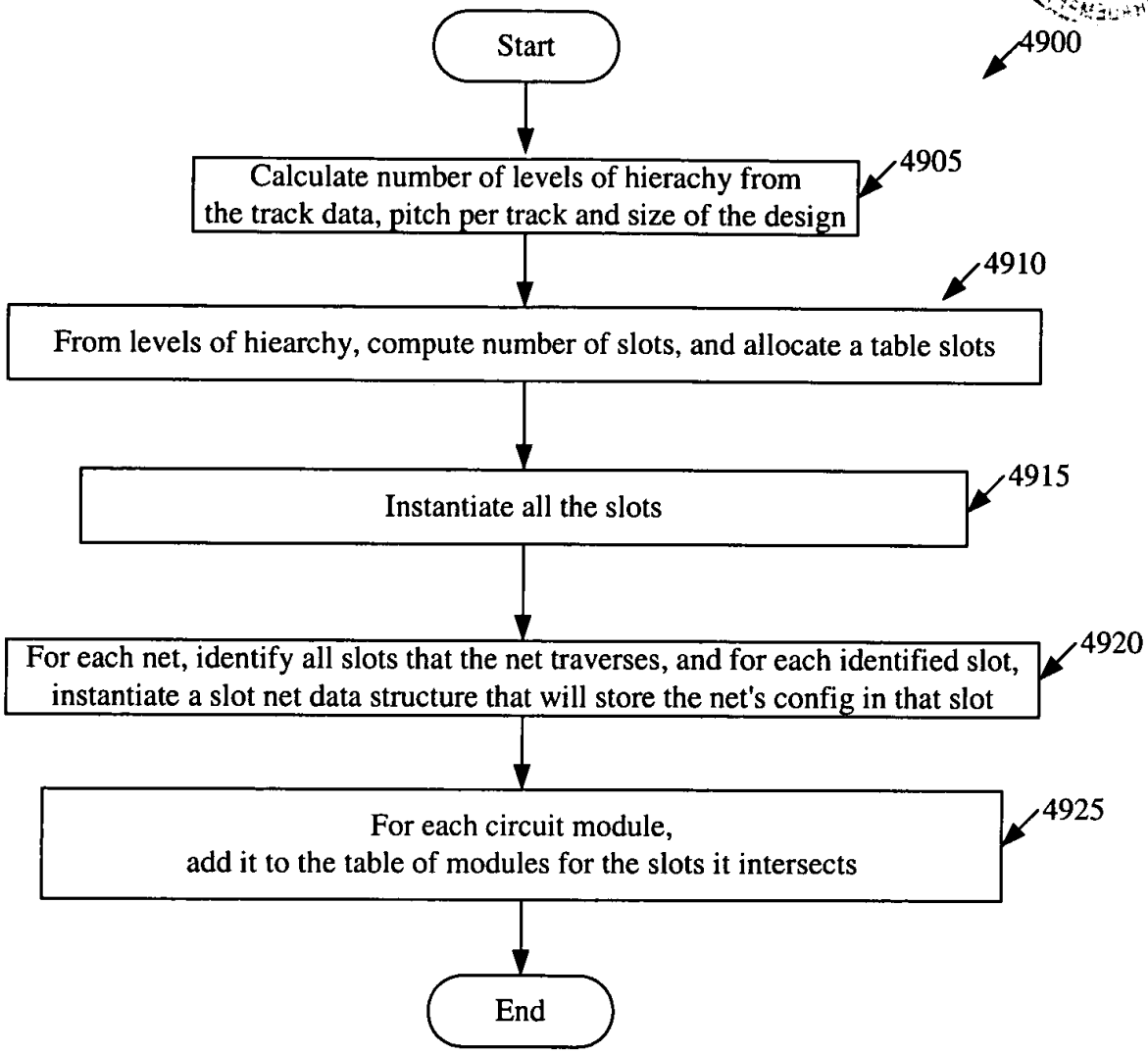

Figure 49

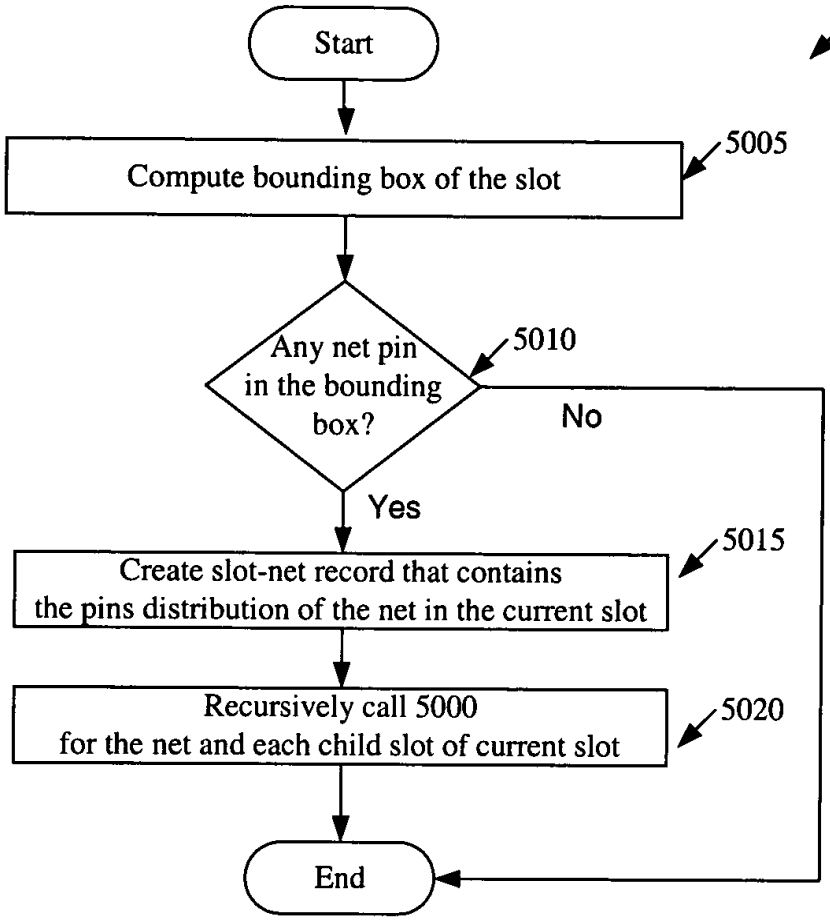
5000
Start
Compute bounding box of the slot
5005
Any net pin in the bounding box?
5010
No
Yes
Create slot-net record that contains the pins distribution of the net in the current slot
5015
Recursively call 5000 for the net and each child slot of current slot
5020
End

Figure 50

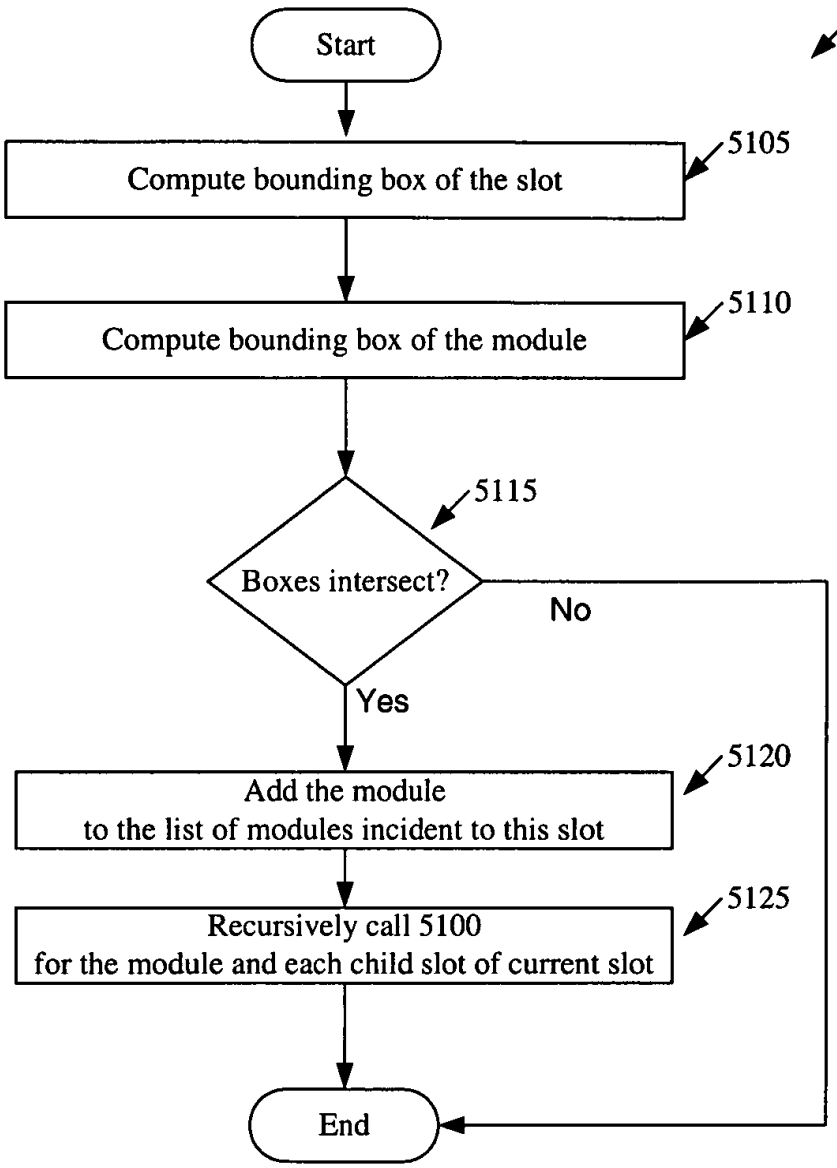
5100
Start
Compute bounding box of the slot
5105
Compute bounding box of the module
5110
5115
Boxes intersect?
No
Yes
Add the module
to the list of modules incident to this slot
5120
Recursively call 5100
for the module and each child slot of current slot
5125
End

Figure 51

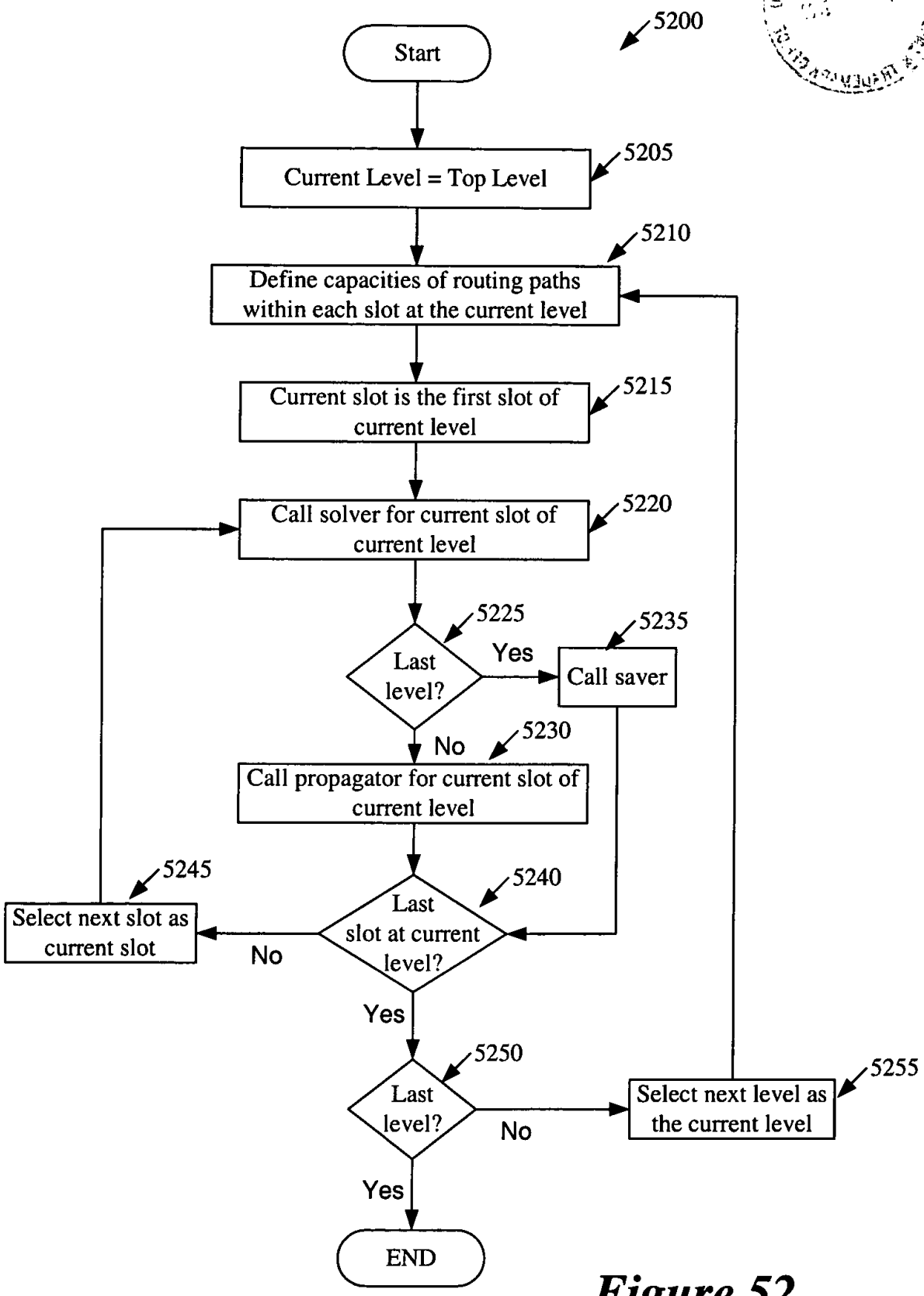
5200
Start
Current Level = Top Level
5205
Define capacities of routing paths within each slot at the current level
5210
Current slot is the first slot of current level
5215
Call solver for current slot of current level
5220
Last level?
5225
Yes
Call saver
5235
No
Call propagator for current slot of current level
5230
Last slot at current level?
5240
No
Select next slot as current slot
5245
Yes
Last level?
5250
No
Select next level as the current level
5255
Yes
END

Figure 52

10079618 .041702

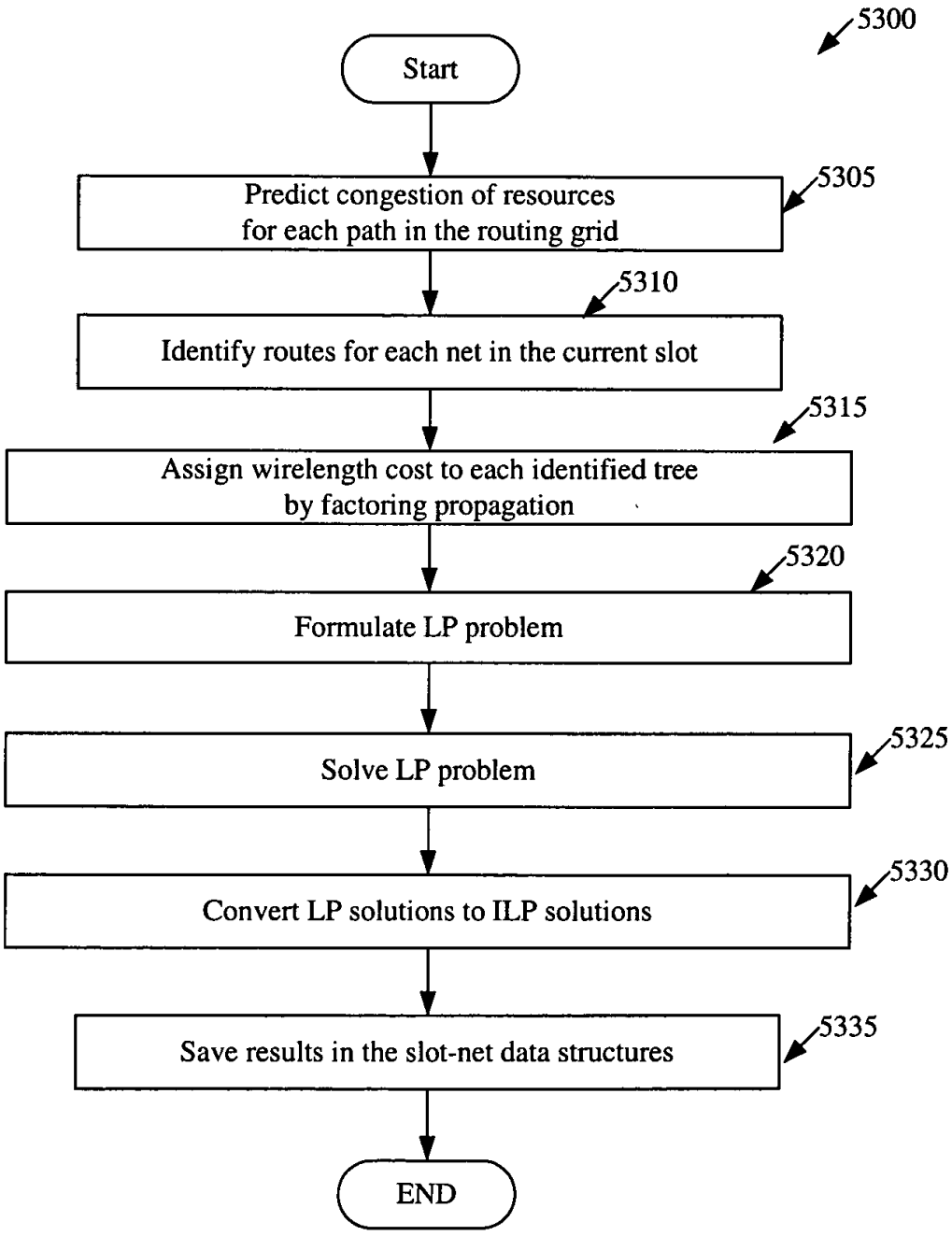

Figure 53

10029618.041702

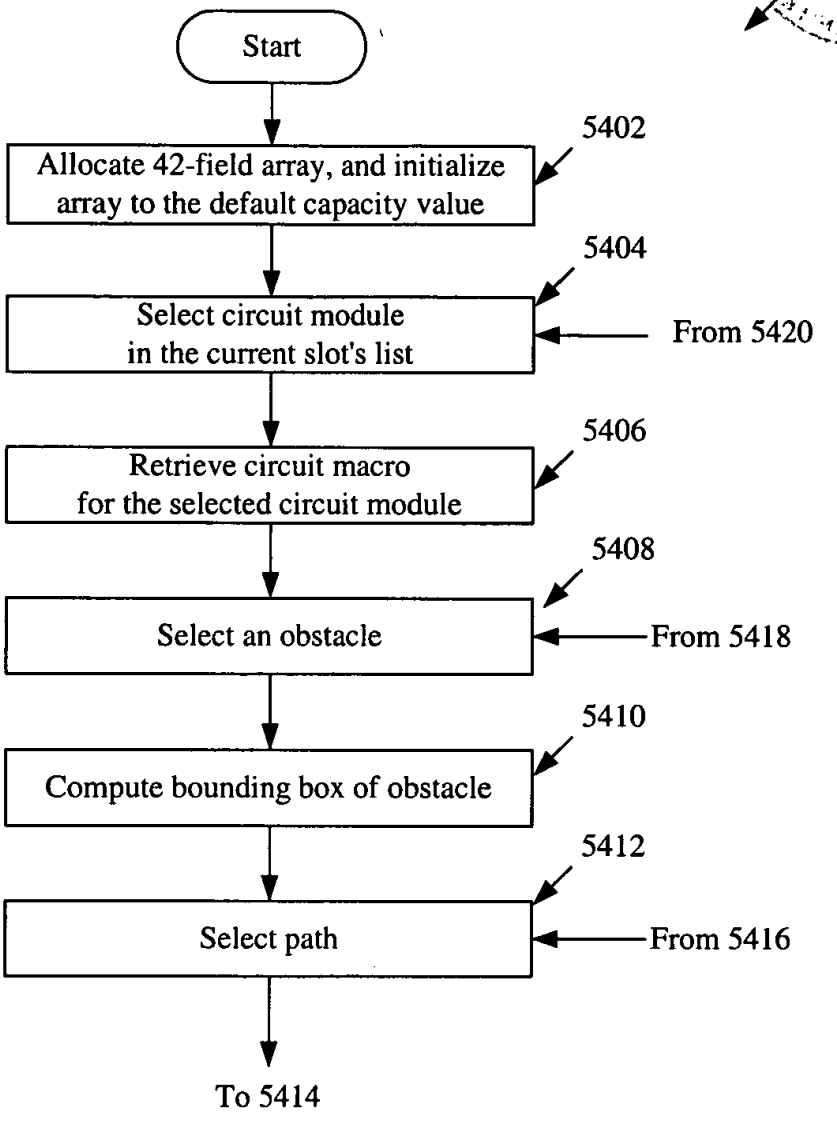

Figure 54A

Figure 54: ***Figure 54A*** / ***Figure 54B***

10029618 .041702

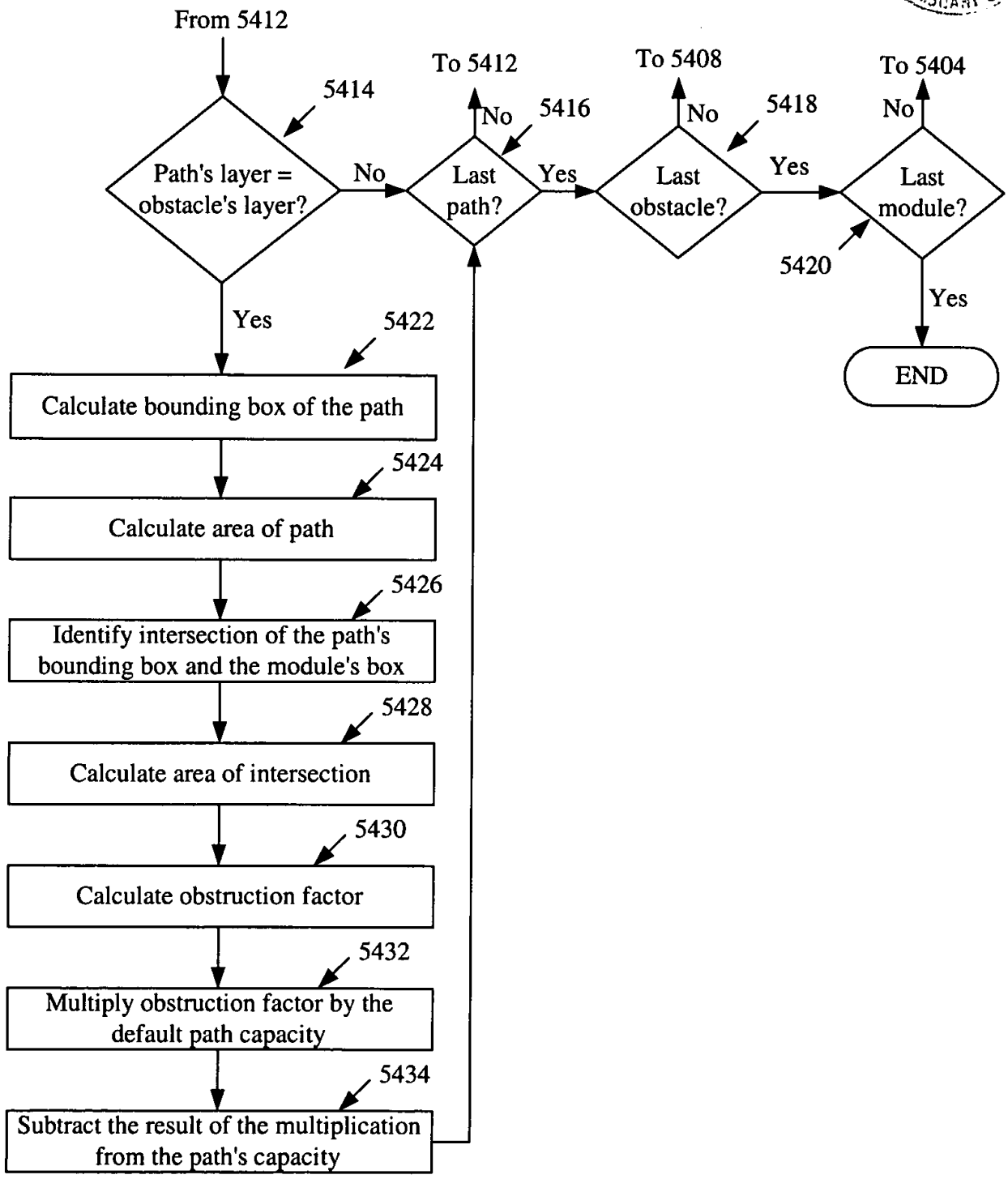

Figure 54B

10029618.041702

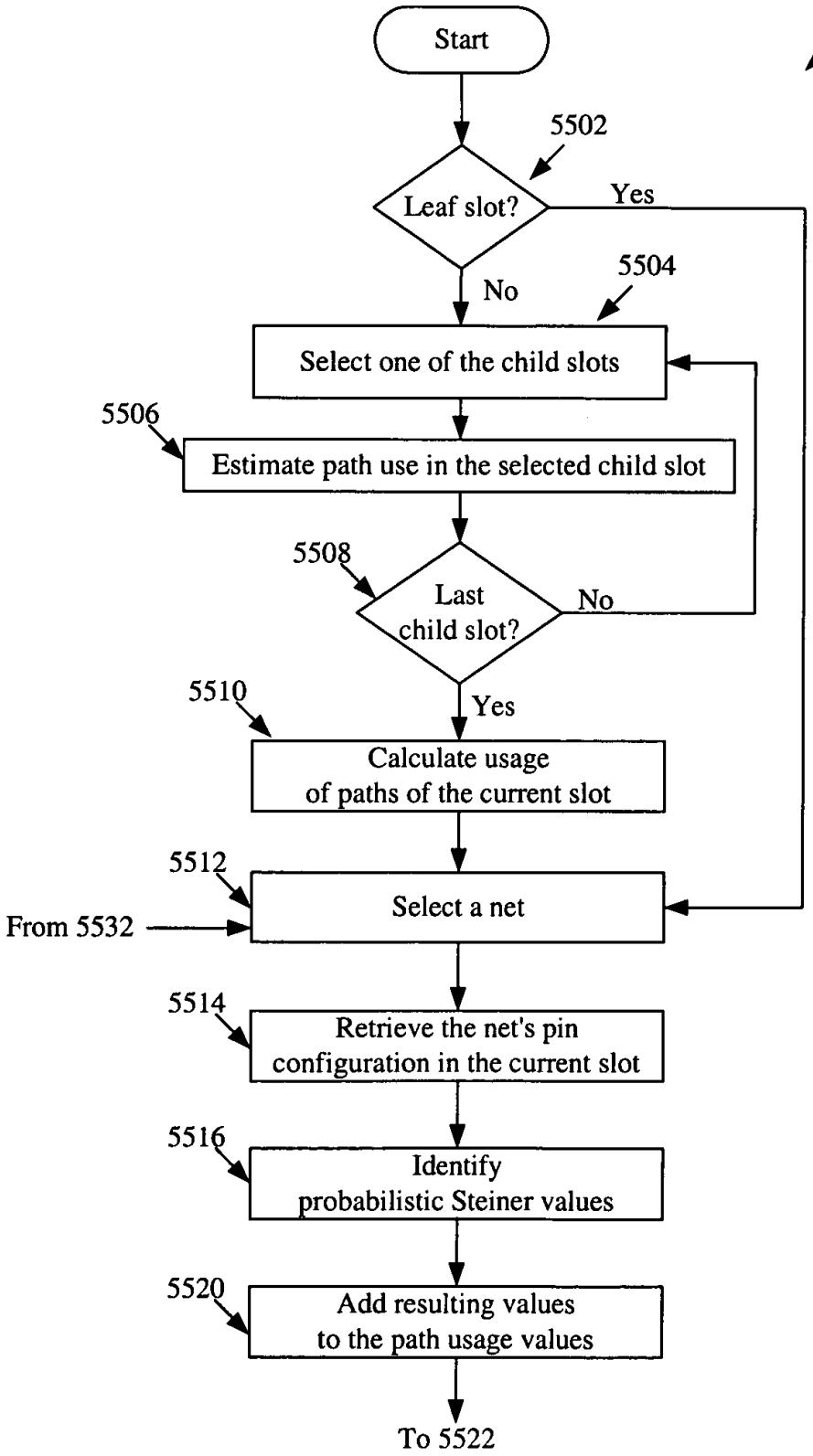

Figure 55A

Figure 55: Figure 55A / Figure 55B

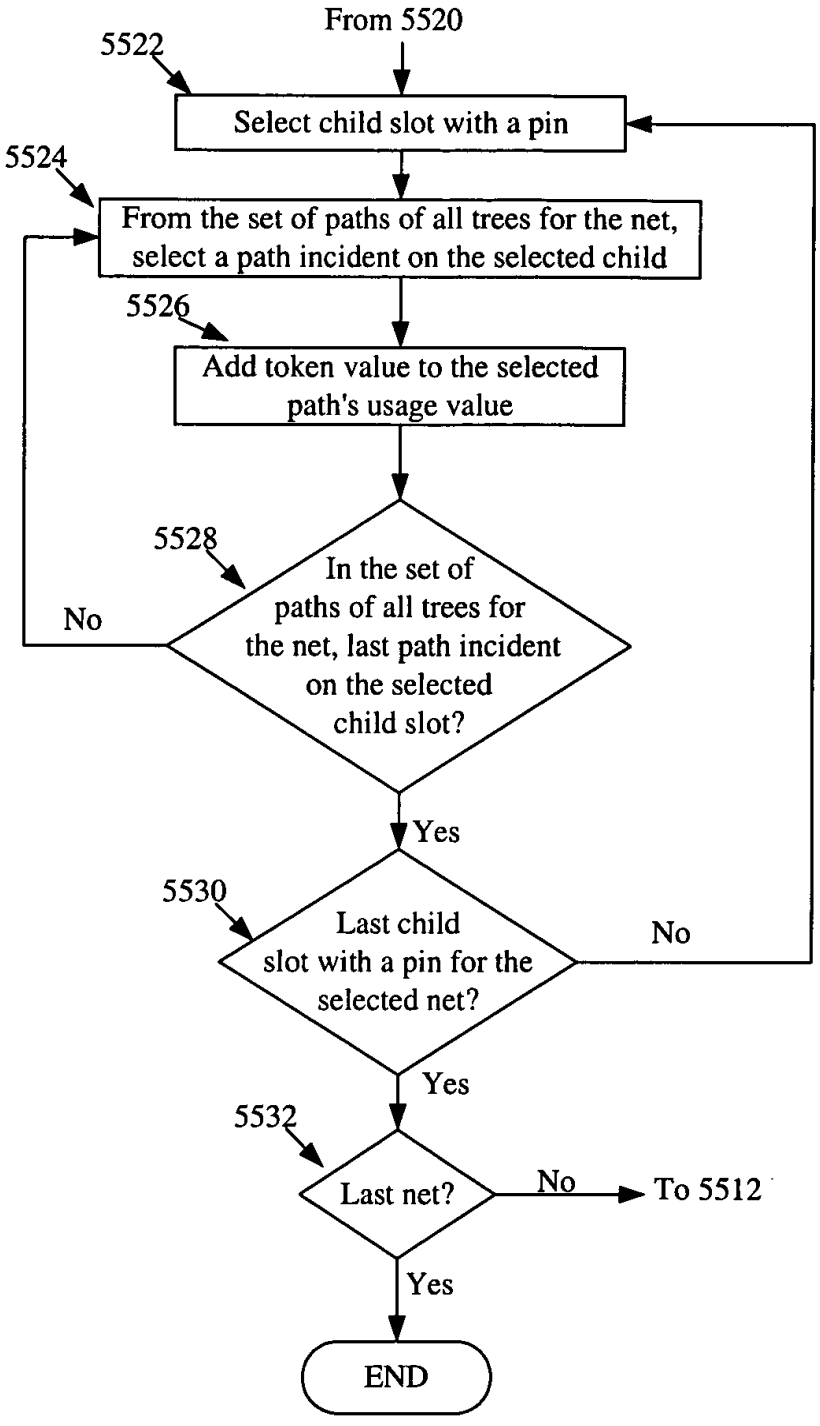
From 5520
5522
Select child slot with a pin
5524
From the set of paths of all trees for the net, select a path incident on the selected child
5526
Add token value to the selected path's usage value
5528
In the set of paths of all trees for the net, last path incident on the selected child slot?
No
Yes
5530
Last child slot with a pin for the selected net?
No
Yes
5532
Last net?
No
To 5512
Yes
END

Figure 55B

10039618.041702

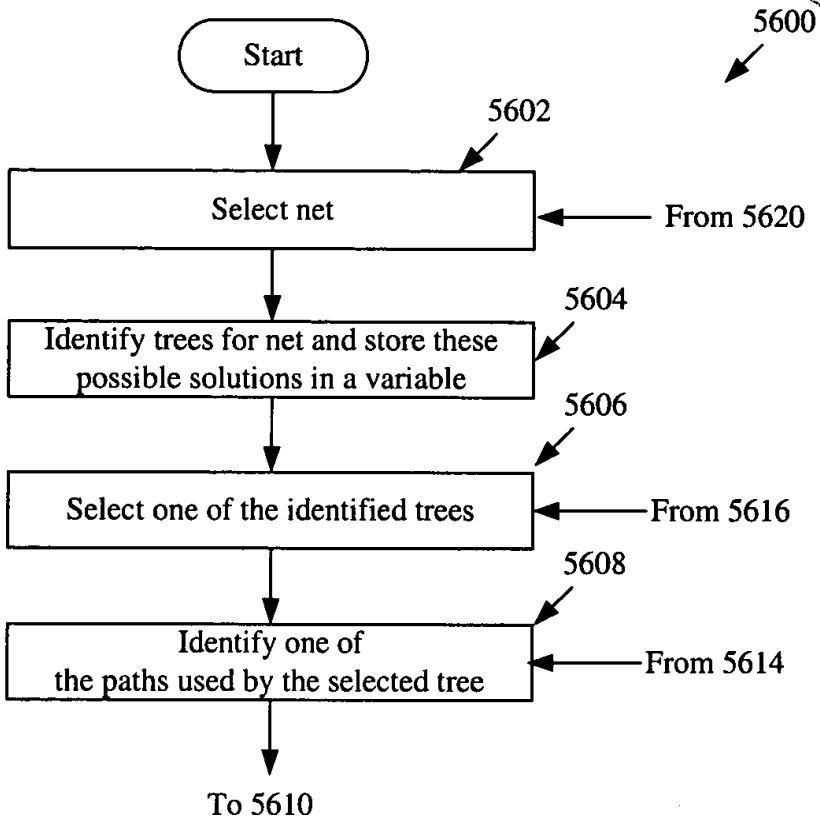

Figure 56A

Figure 56: *Figure 56A* / *Figure 56B* / *Figure 56C* / *Figure 56D*

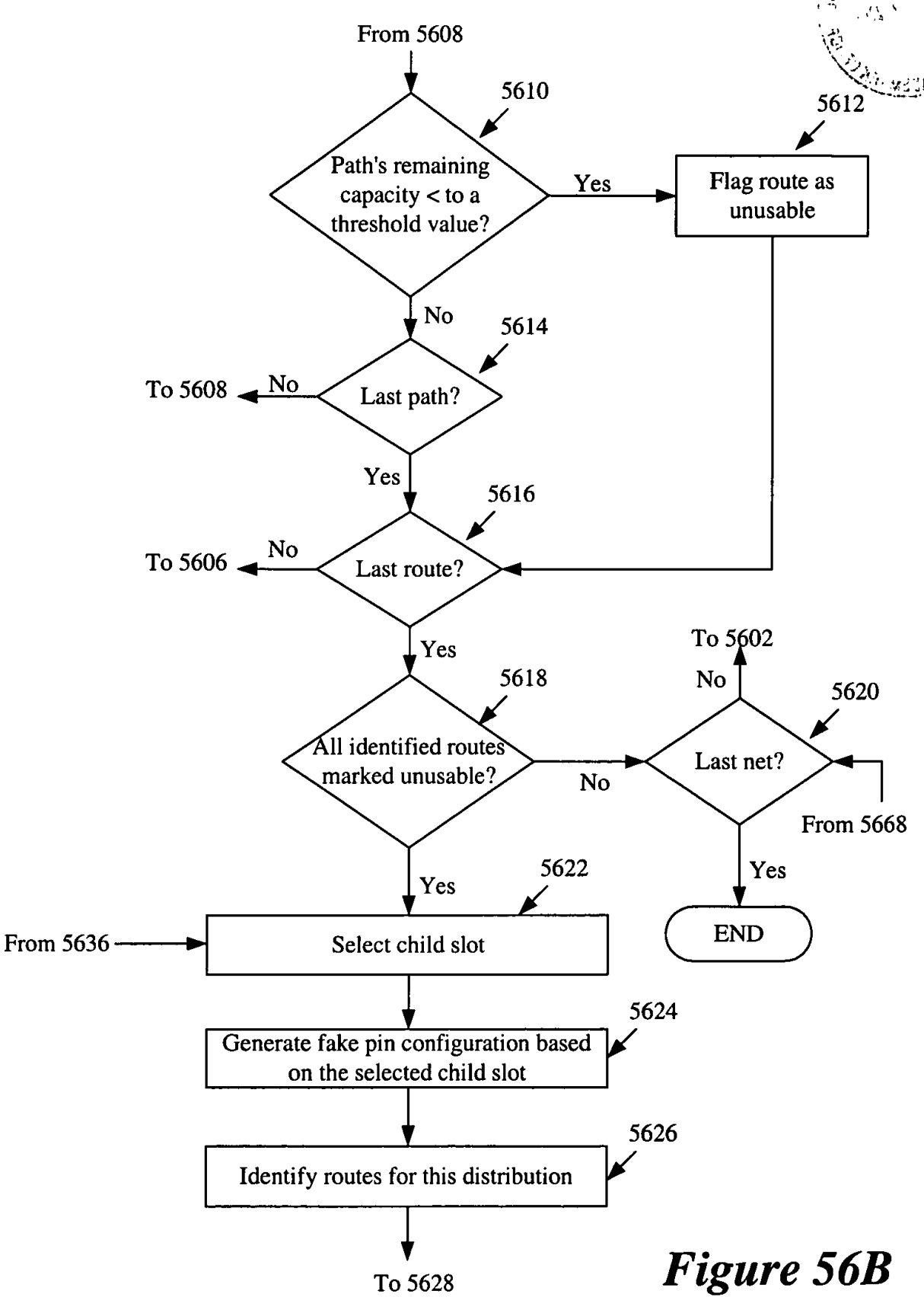
From 5608
5610
Path's remaining capacity < to a threshold value?
Yes
5612
Flag route as unusable
No
5614
To 5608
No
Last path?
Yes
5616
To 5606
No
Last route?
Yes
To 5602
No
5618
All identified routes marked unusable?
No
5620
Last net?
From 5668
Yes
Yes
5622
END
From 5636
Select child slot
5624
Generate fake pin configuration based on the selected child slot
5626
Identify routes for this distribution
To 5628

Figure 56B

10029618.041702

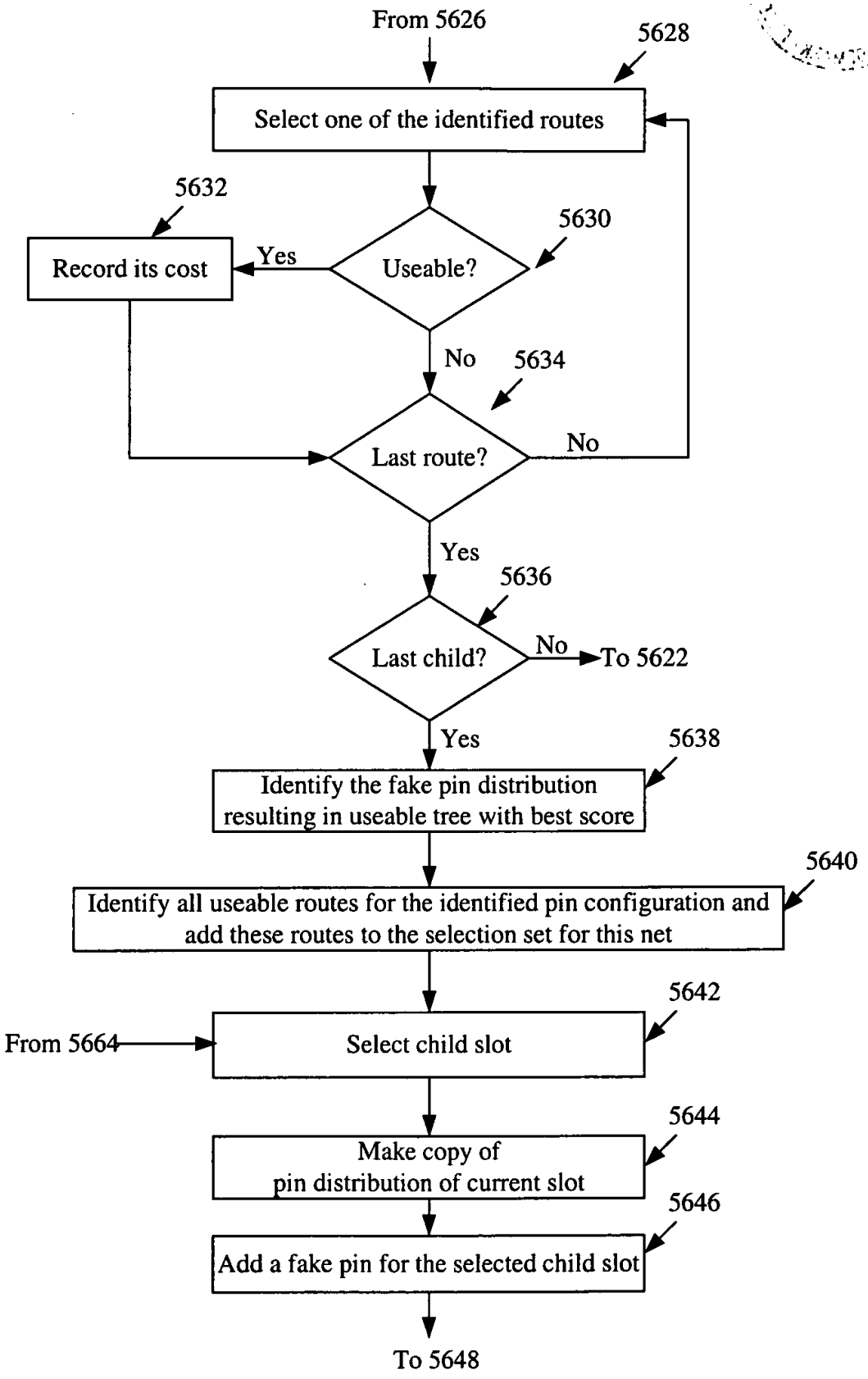

Figure 56C

10079618.041702

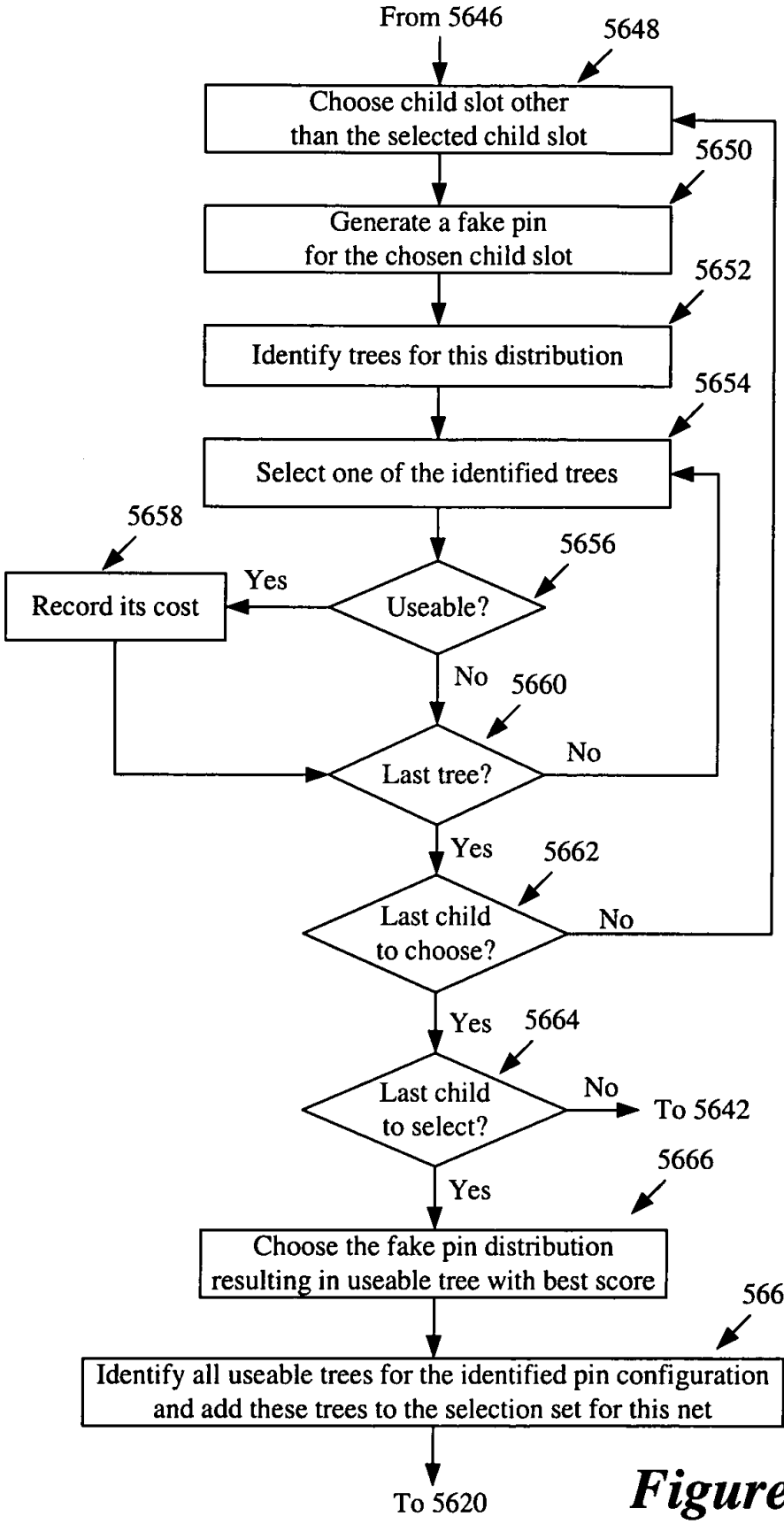

Figure 56D

10079618 .041702

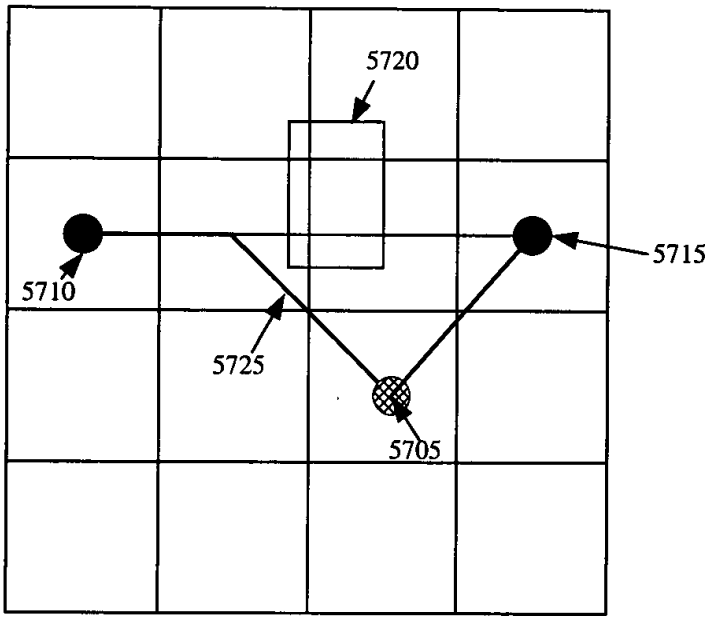

Figure 57

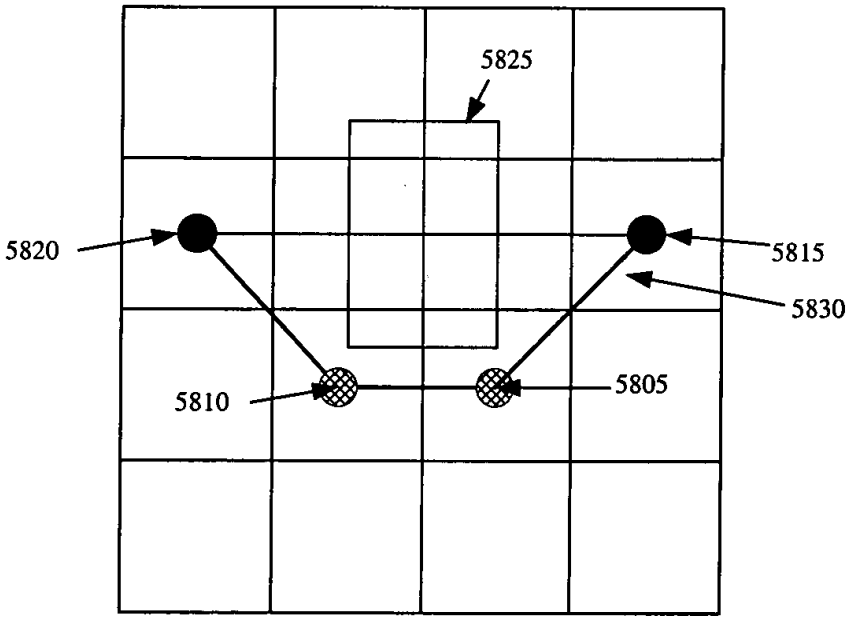

Figure 58

10079618 .041702

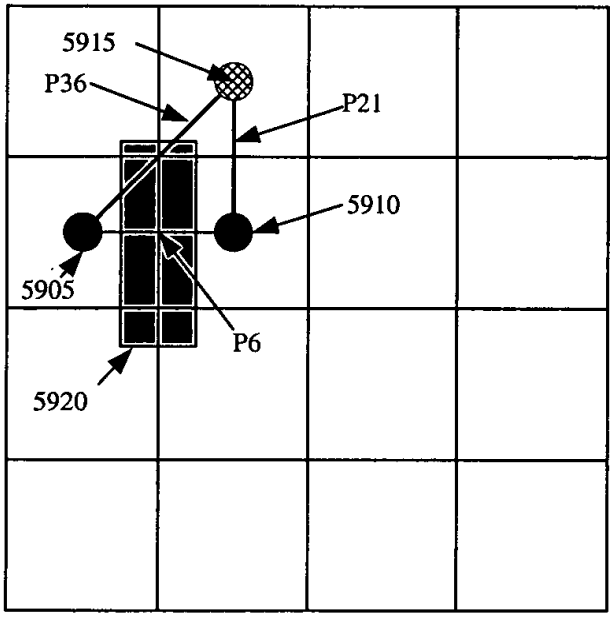

Figure 59

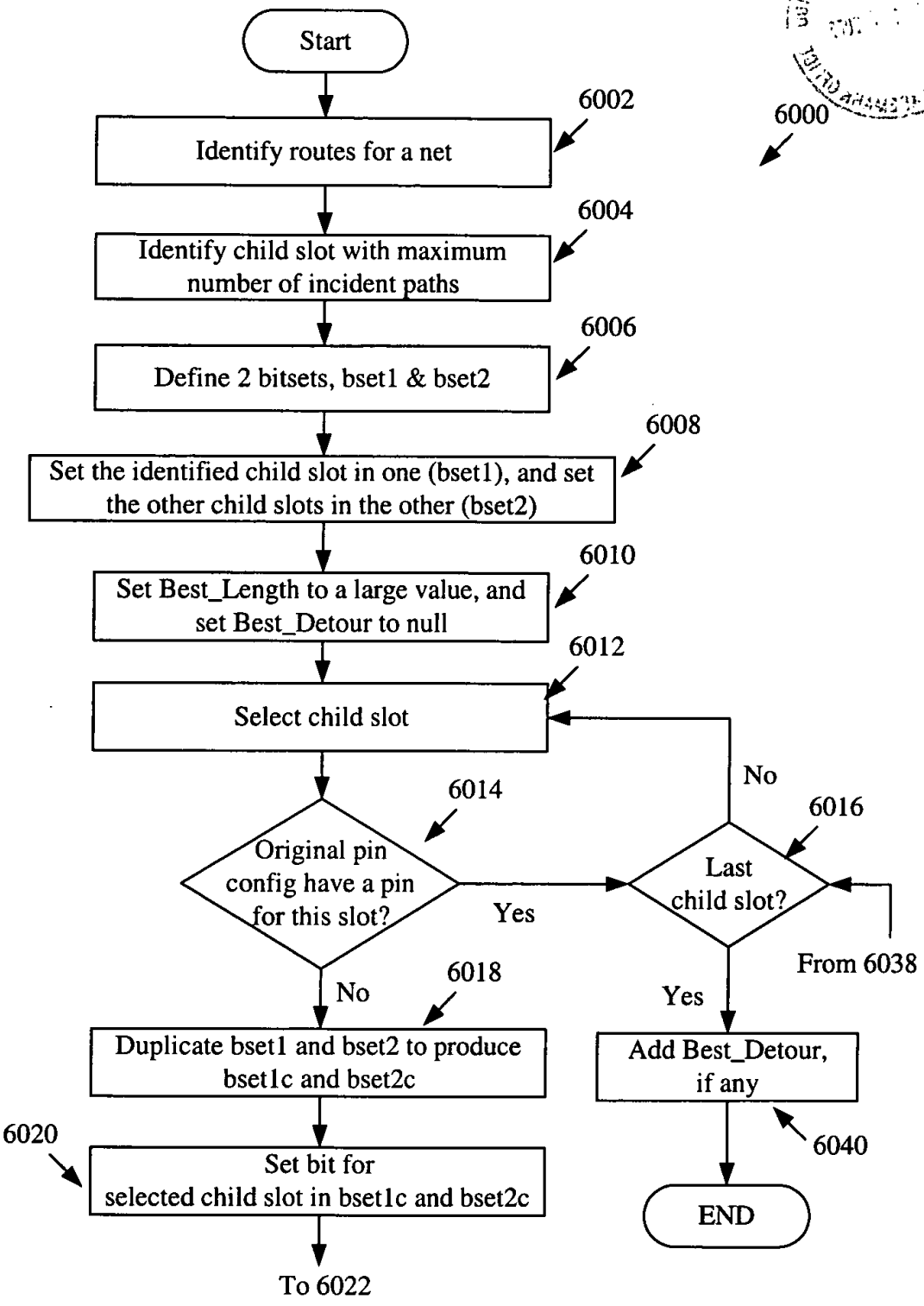

Figure 60A

Figure 60: Figure 60A / Figure 60B

10079618 .041702

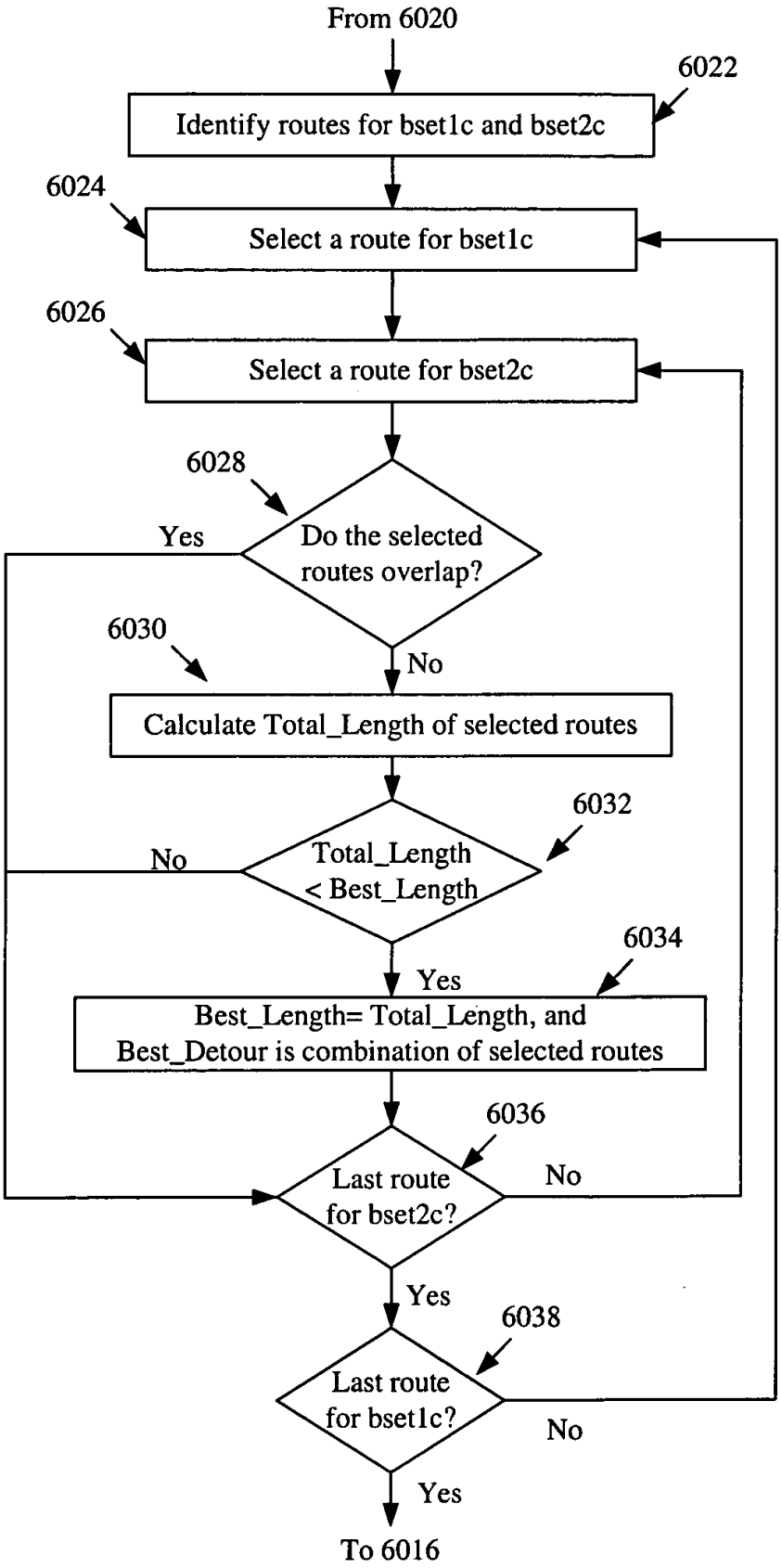

Figure 60B

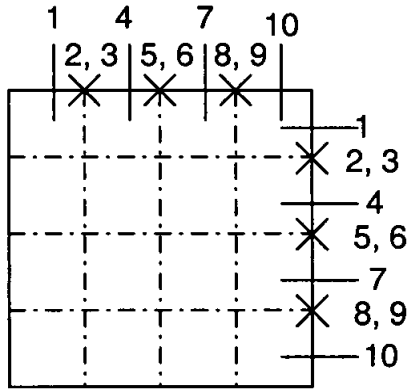

Figure 61

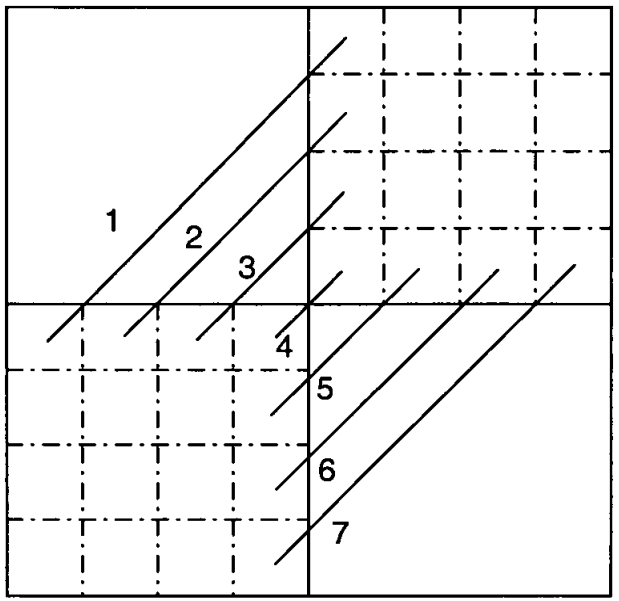

Figure 62

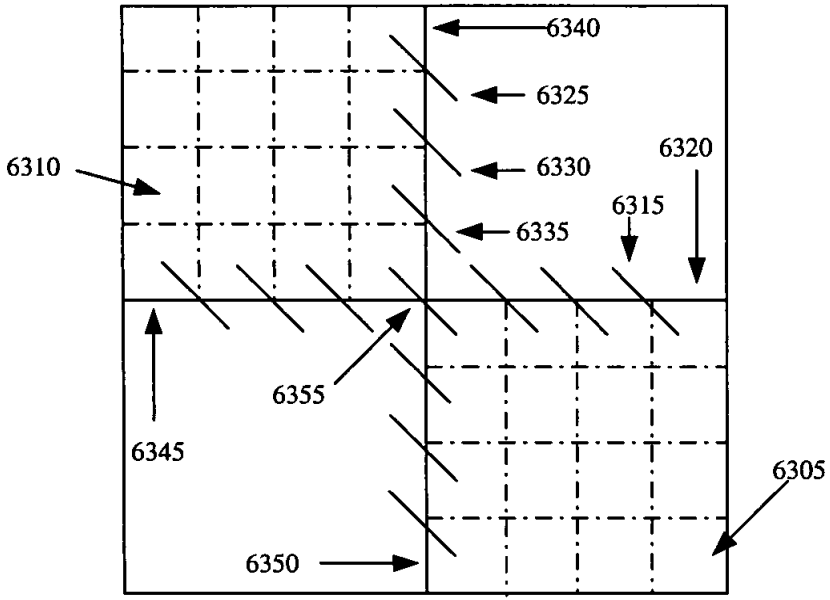

Figure 63

20070418.0417702

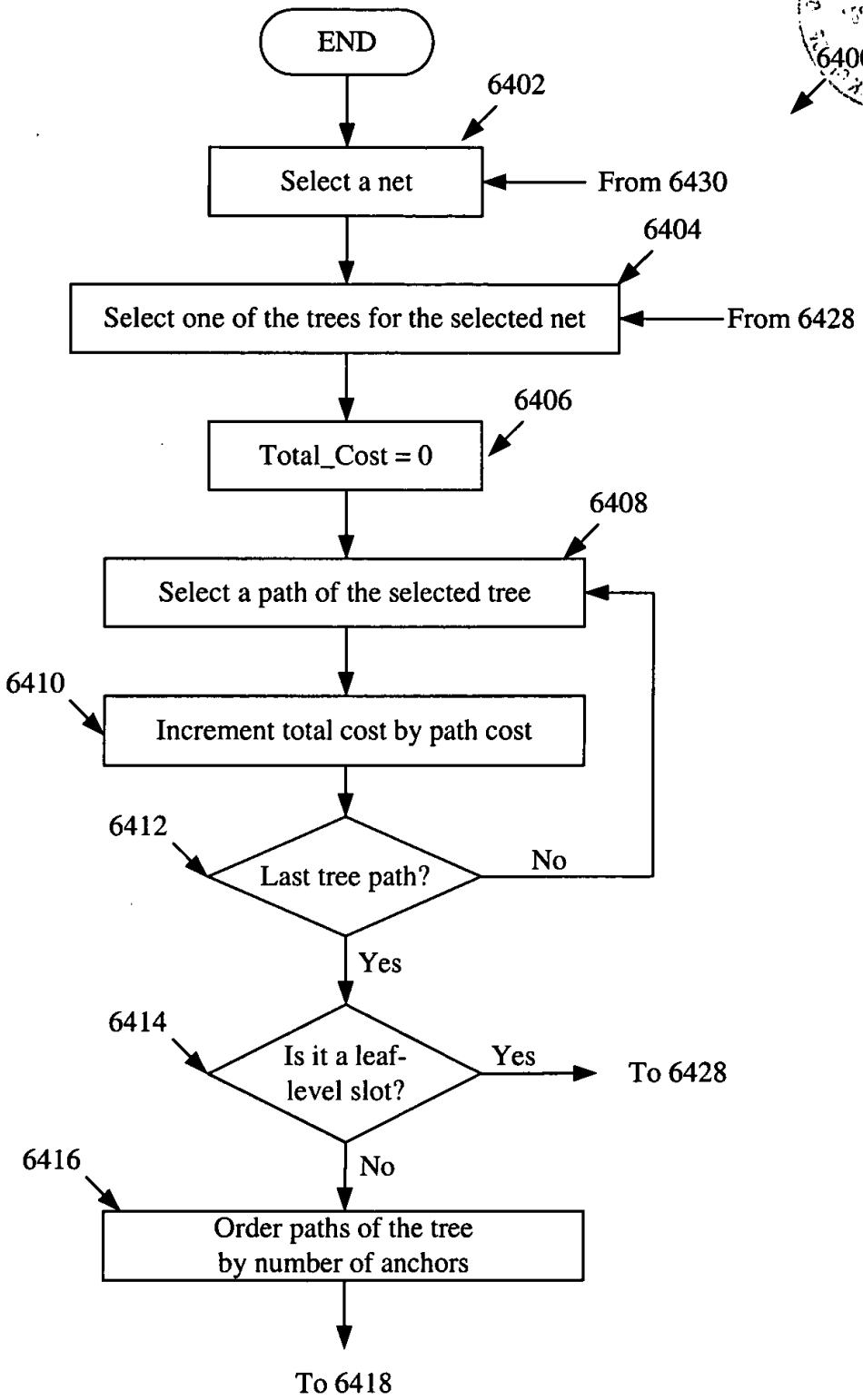

Figure 64A

Figure 64: Figure 64A / Figure 64B

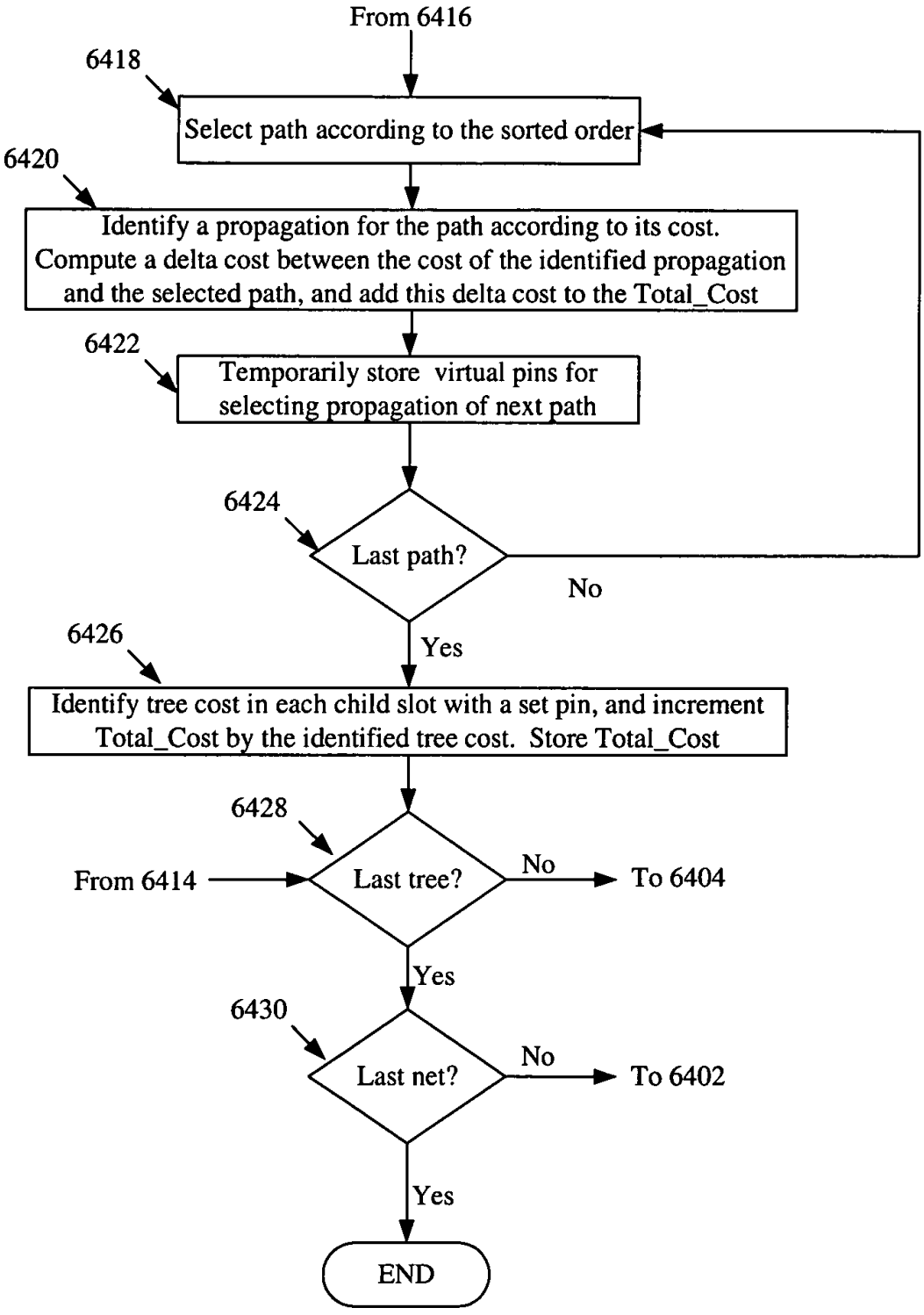
From 6416
6418
Select path according to the sorted order
6420
Identify a propagation for the path according to its cost.
Compute a delta cost between the cost of the identified propagation
and the selected path, and add this delta cost to the Total_Cost
6422
Temporarily store virtual pins for
selecting propagation of next path
6424
Last path?
No
Yes
6426
Identify tree cost in each child slot with a set pin, and increment
Total_Cost by the identified tree cost. Store Total_Cost
6428
From 6414
Last tree?
No
To 6404
Yes
6430
Last net?
No
To 6402
Yes
END

Figure 64B

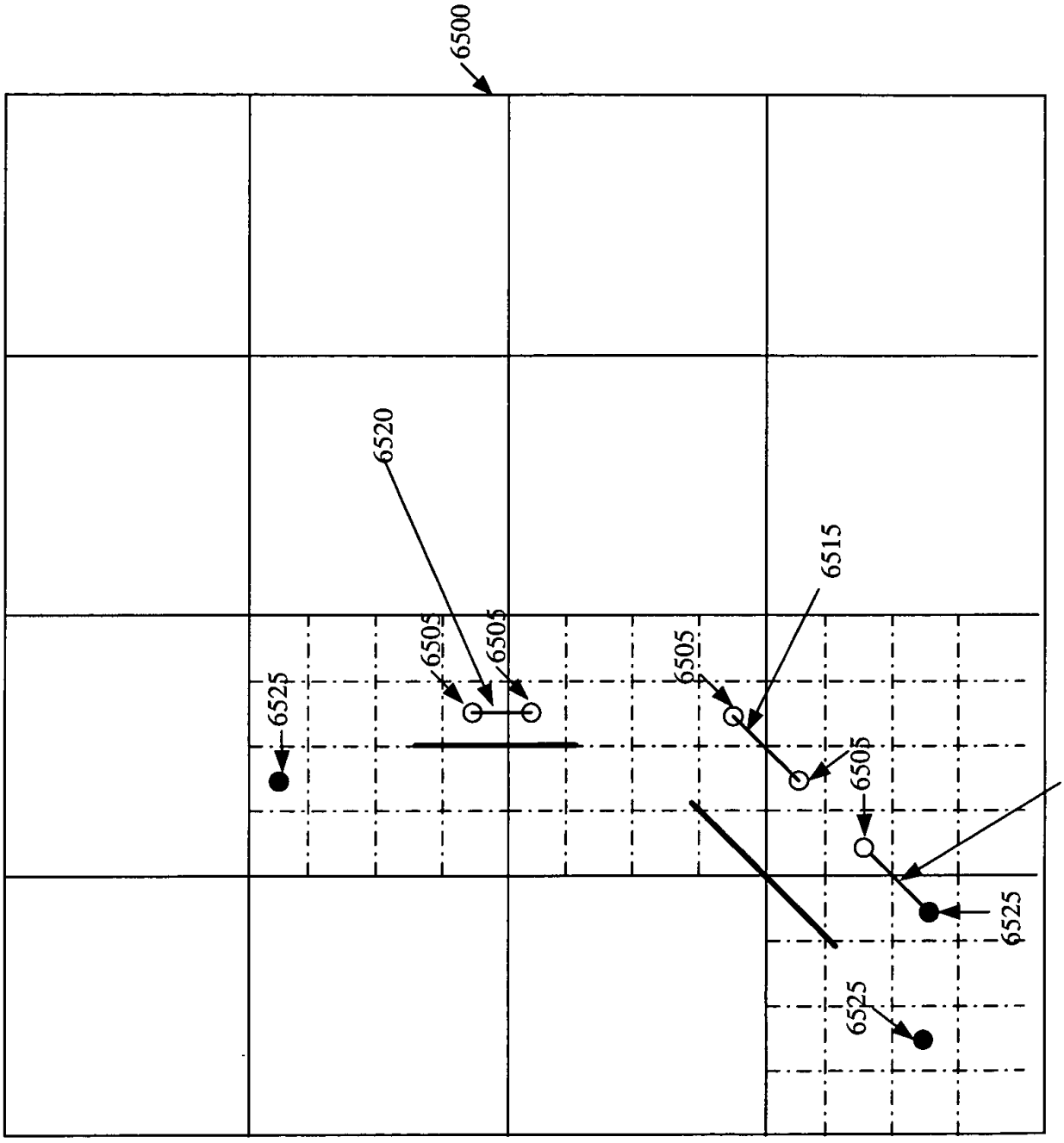

Figure 65

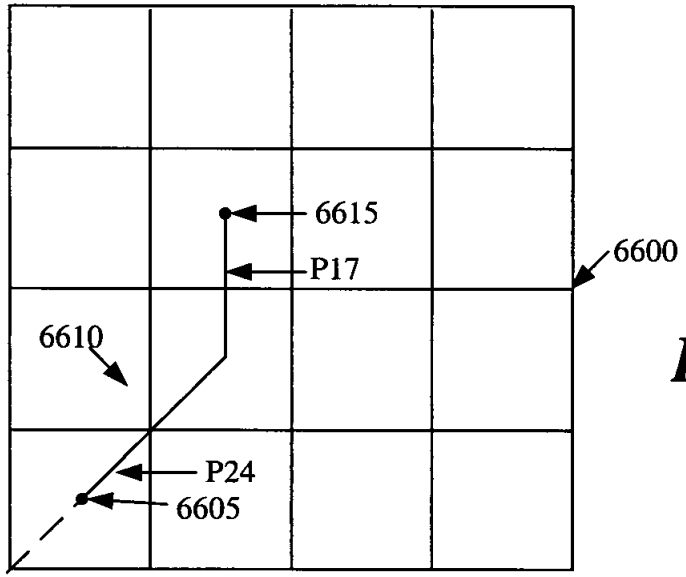

Figure 66

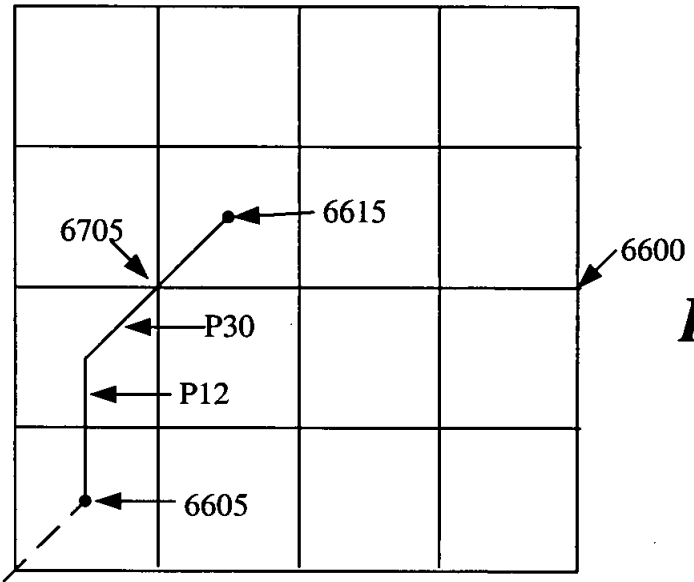

Figure 67

20070618.0417.02

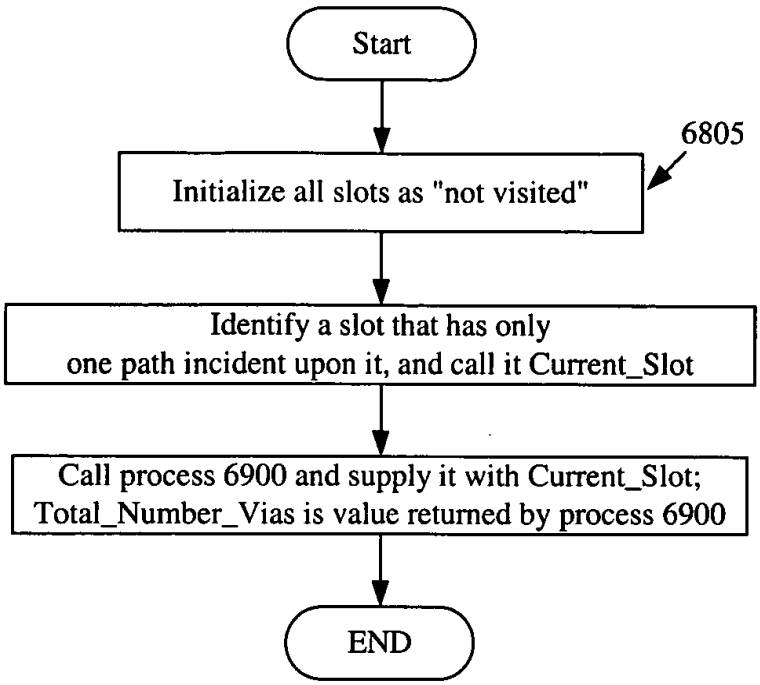

Figure 68

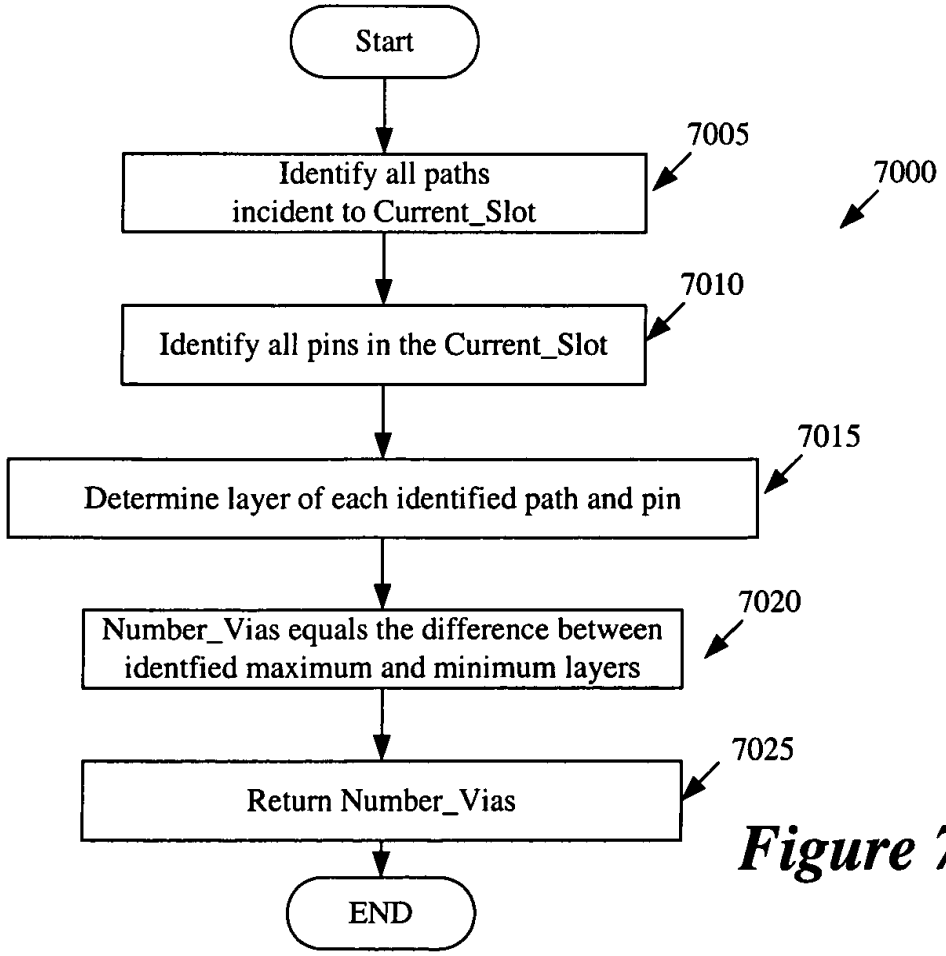

Figure 70

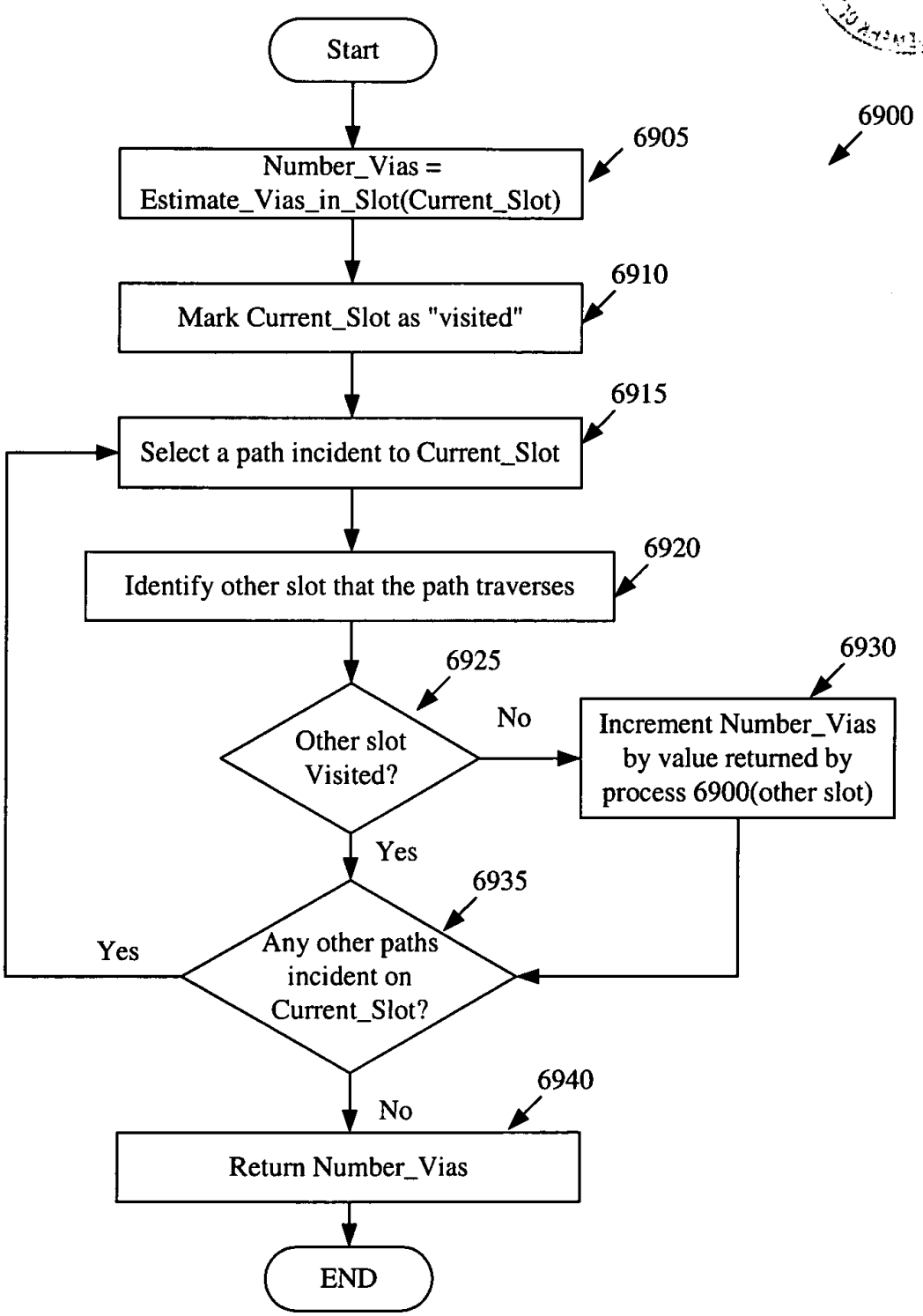
Start
6900
6905
Number_Vias = Estimate_Vias_in_Slot(Current_Slot)
6910
Mark Current_Slot as "visited"
6915
Select a path incident to Current_Slot
6920
Identify other slot that the path traverses
6925
Other slot Visited?
No
6930
Increment Number_Vias by value returned by process 6900(other slot)
Yes
6935
Any other paths incident on Current_Slot?
Yes
No
6940
Return Number_Vias
END

Figure 69

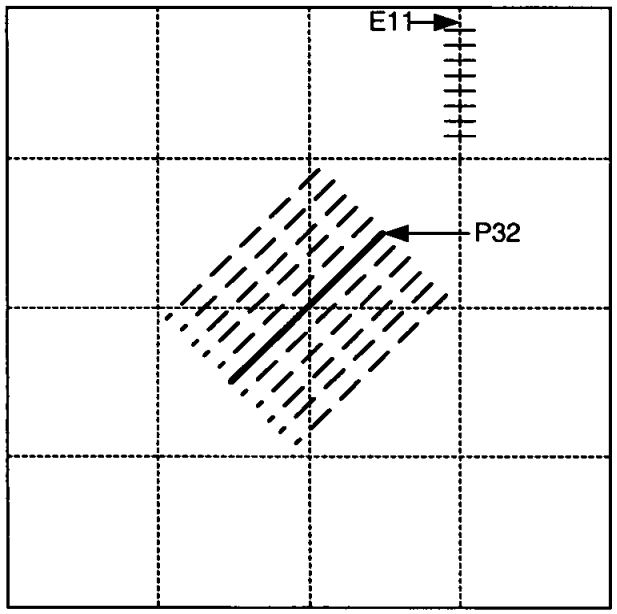

Figure 71

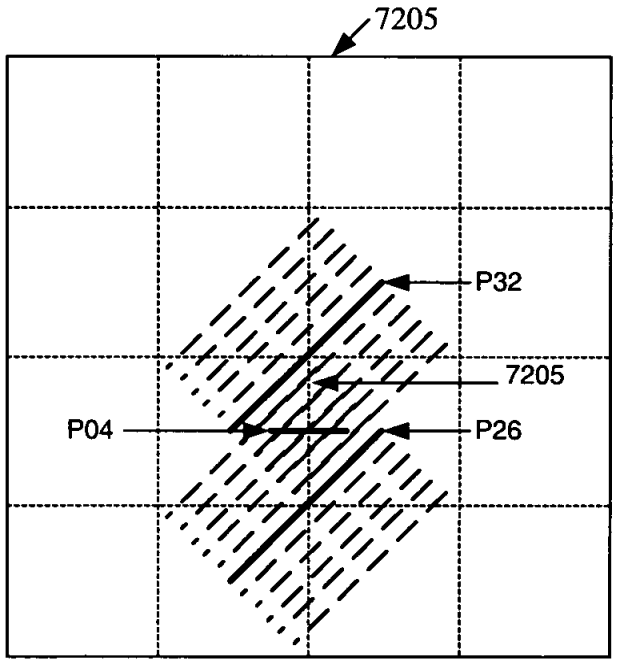

Figure 72

20/4/18, 04:17:02

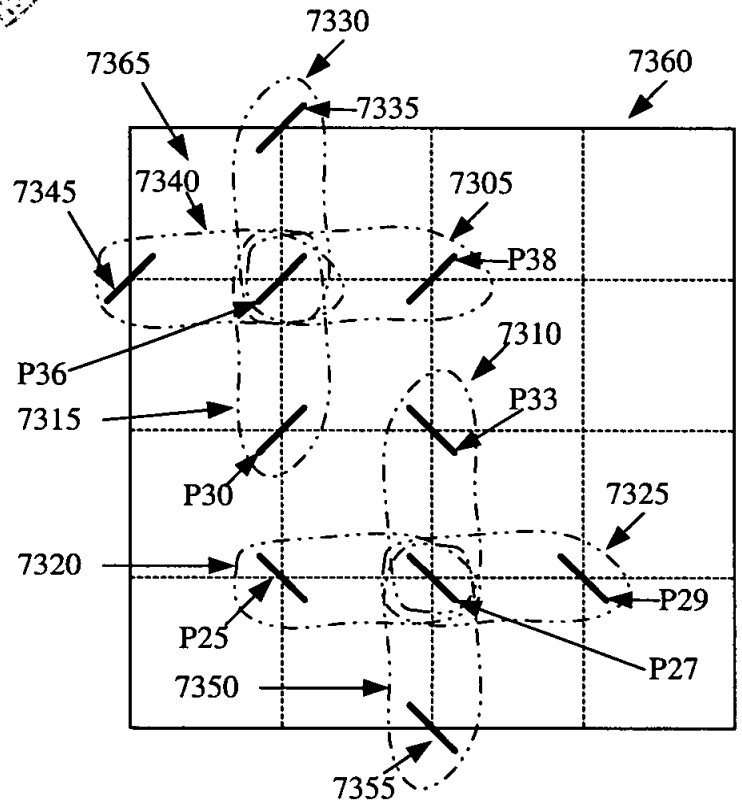

Figure 73

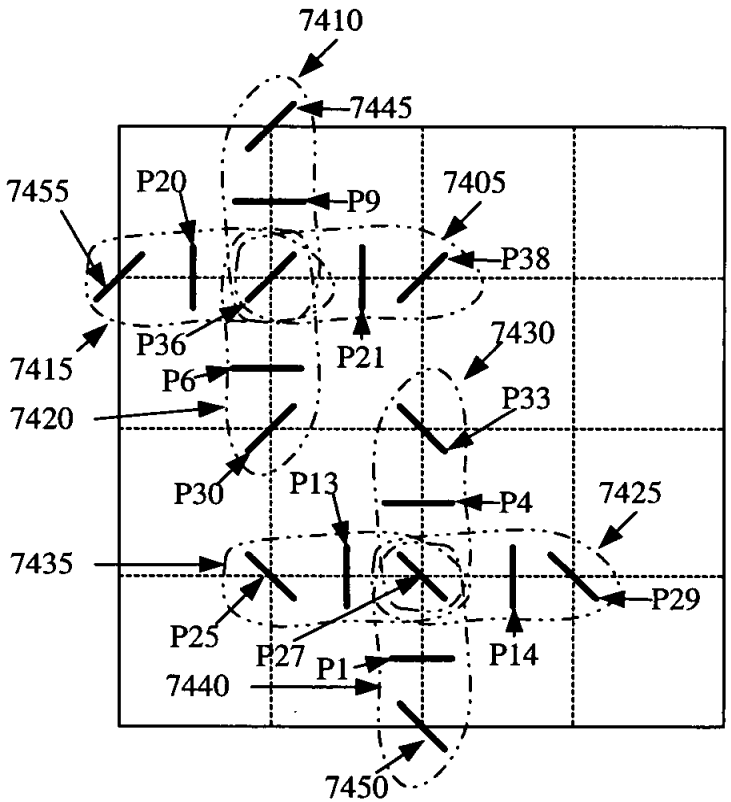

Figure 74

10076618 - 041702

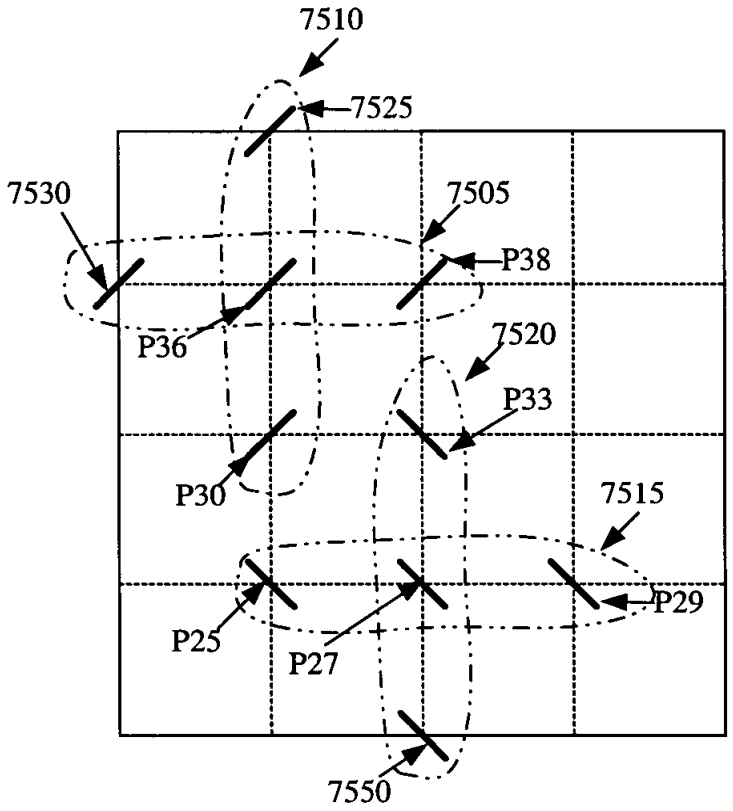

Figure 75

10079618 .041702

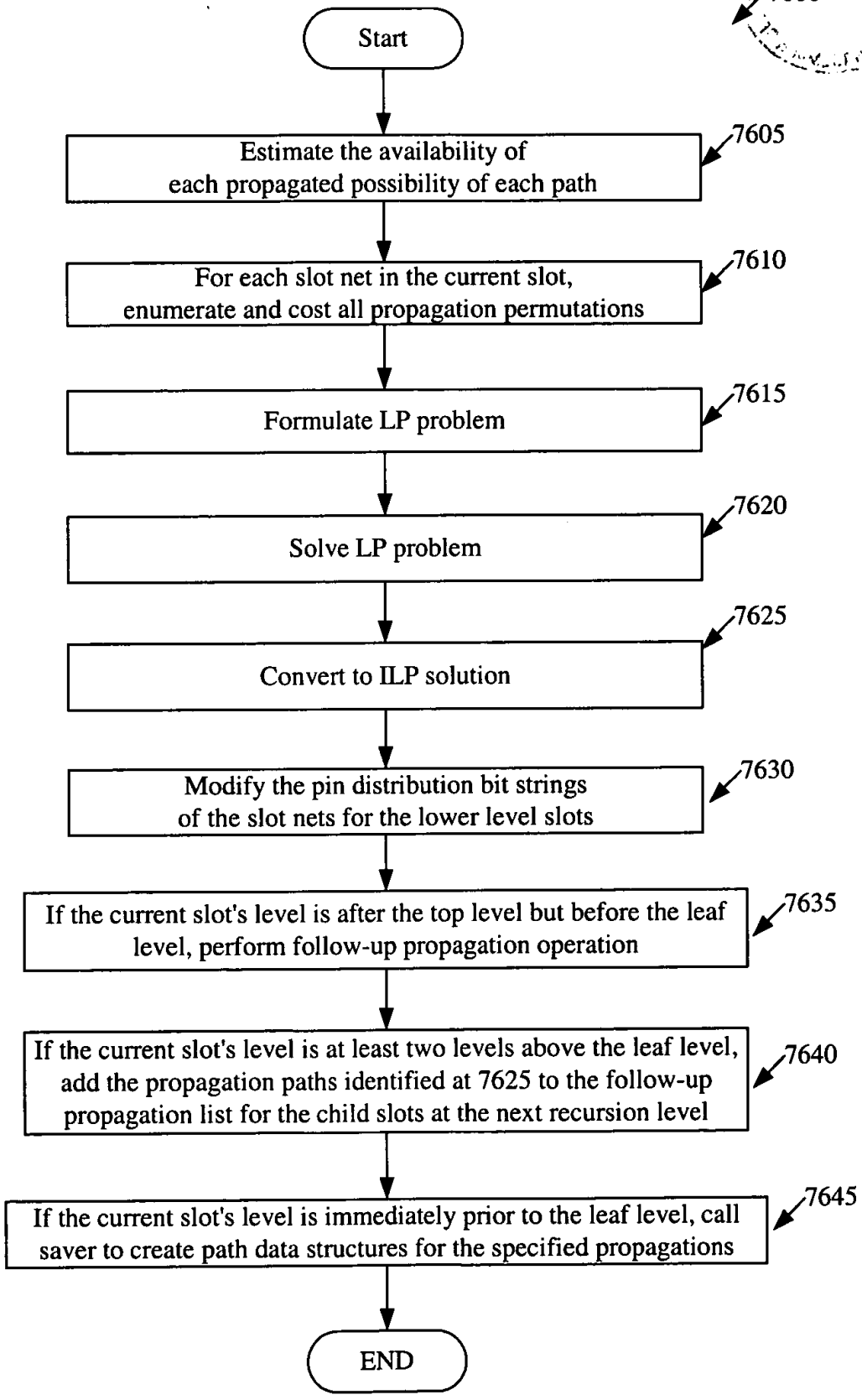

Figure 76

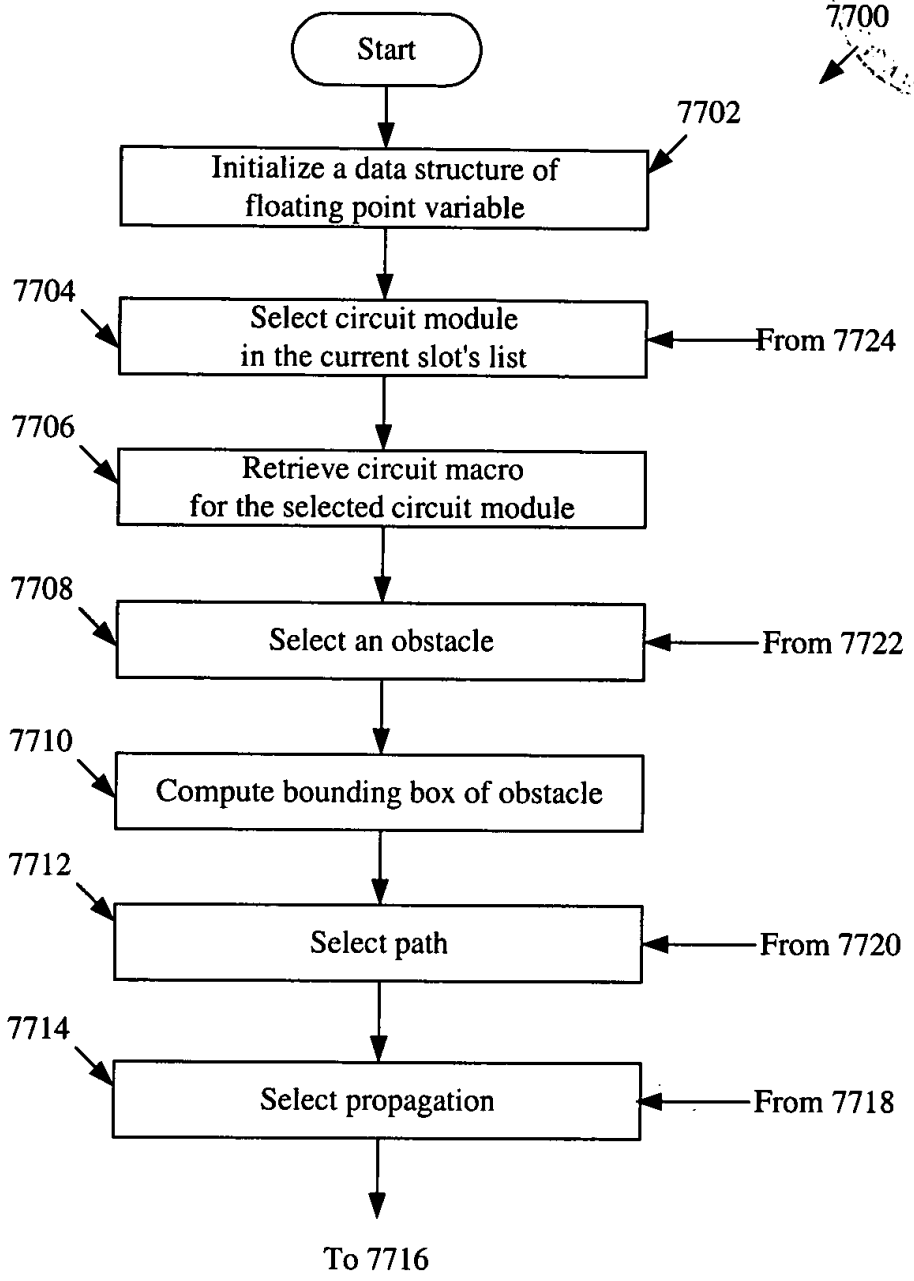

Figure 77A

Figure 77: ***Figure 77A*** / ***Figure 77B***

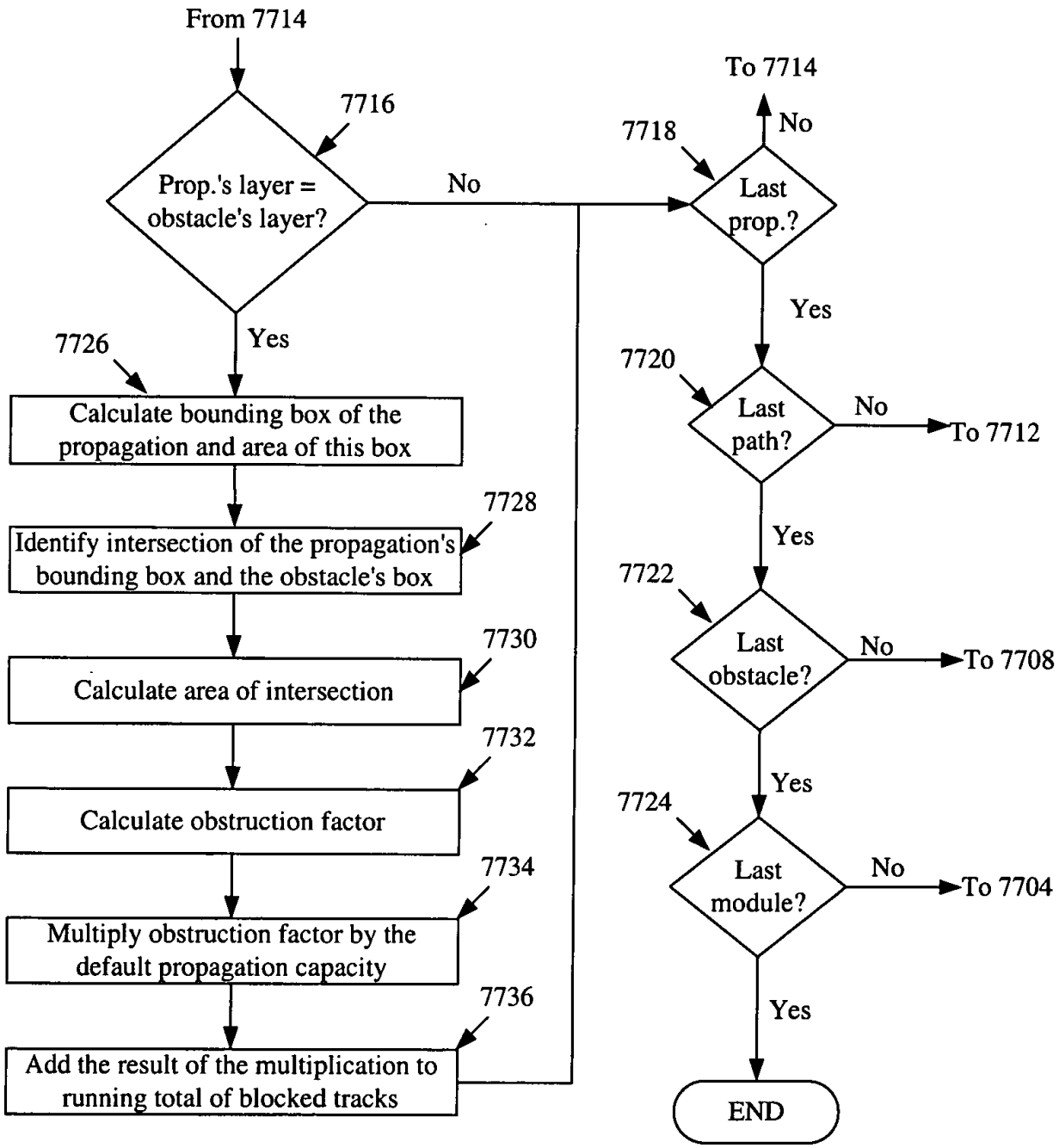
From 7714
7716
Prop.'s layer = obstacle's layer?
No
Yes
7726
Calculate bounding box of the propagation and area of this box
7728
Identify intersection of the propagation's bounding box and the obstacle's box
7730
Calculate area of intersection
7732
Calculate obstruction factor
7734
Multiply obstruction factor by the default propagation capacity
7736
Add the result of the multiplication to running total of blocked tracks
To 7714
7718
No
Last prop.?
Yes
7720
Last path?
No
To 7712
Yes
7722
Last obstacle?
No
To 7708
Yes
7724
Last module?
No
To 7704
Yes
END

Figure 77B

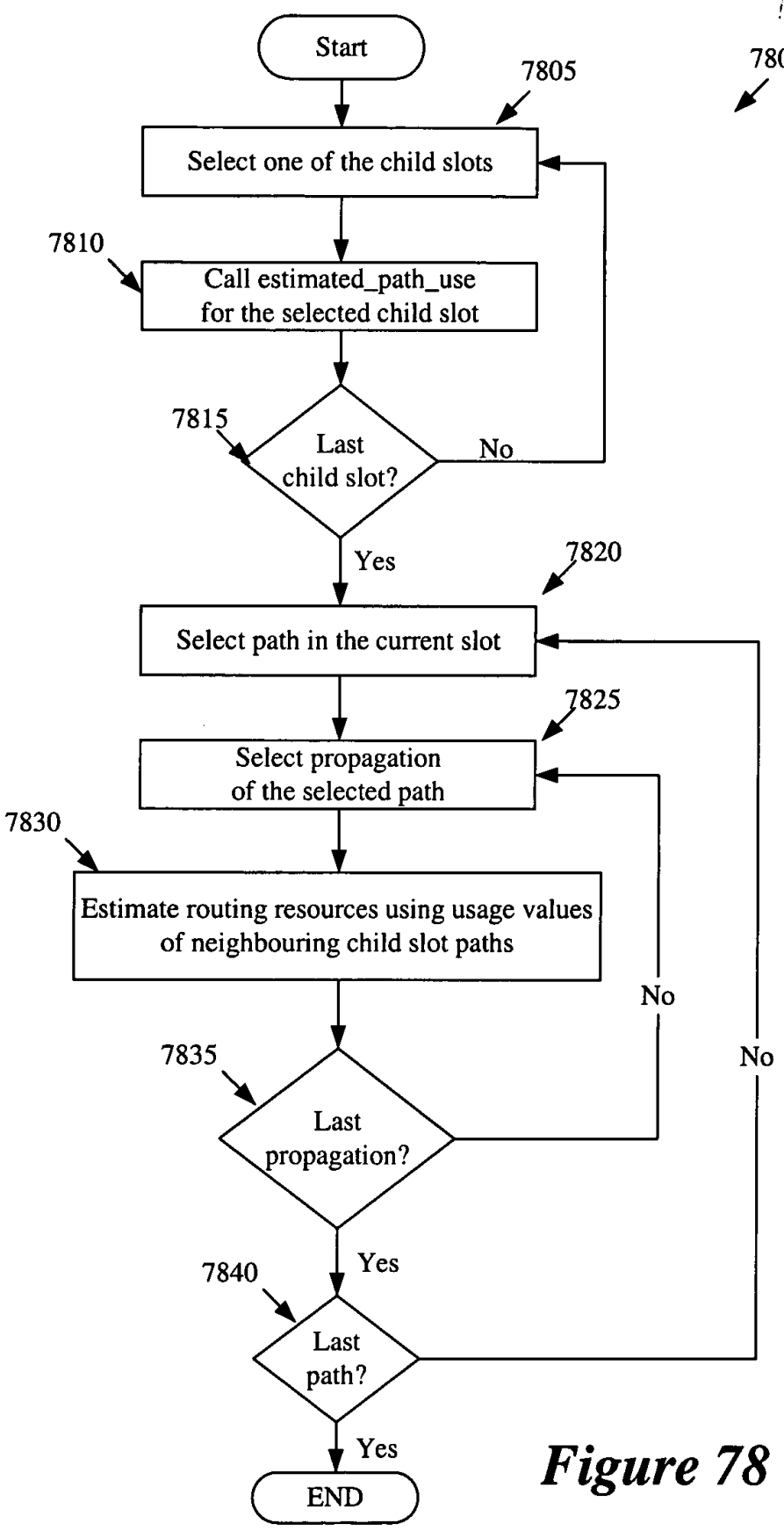
Start
7805
7800
Select one of the child slots
7810
Call estimated_path_use
for the selected child slot
7815
Last
child slot?
No
Yes
7820
Select path in the current slot
7825
Select propagation
of the selected path
7830
Estimate routing resources using usage values
of neighbouring child slot paths
No
7835
Last
propagation?
No
Yes
7840
Last
path?
Yes
END

Figure 78

10079618.041702

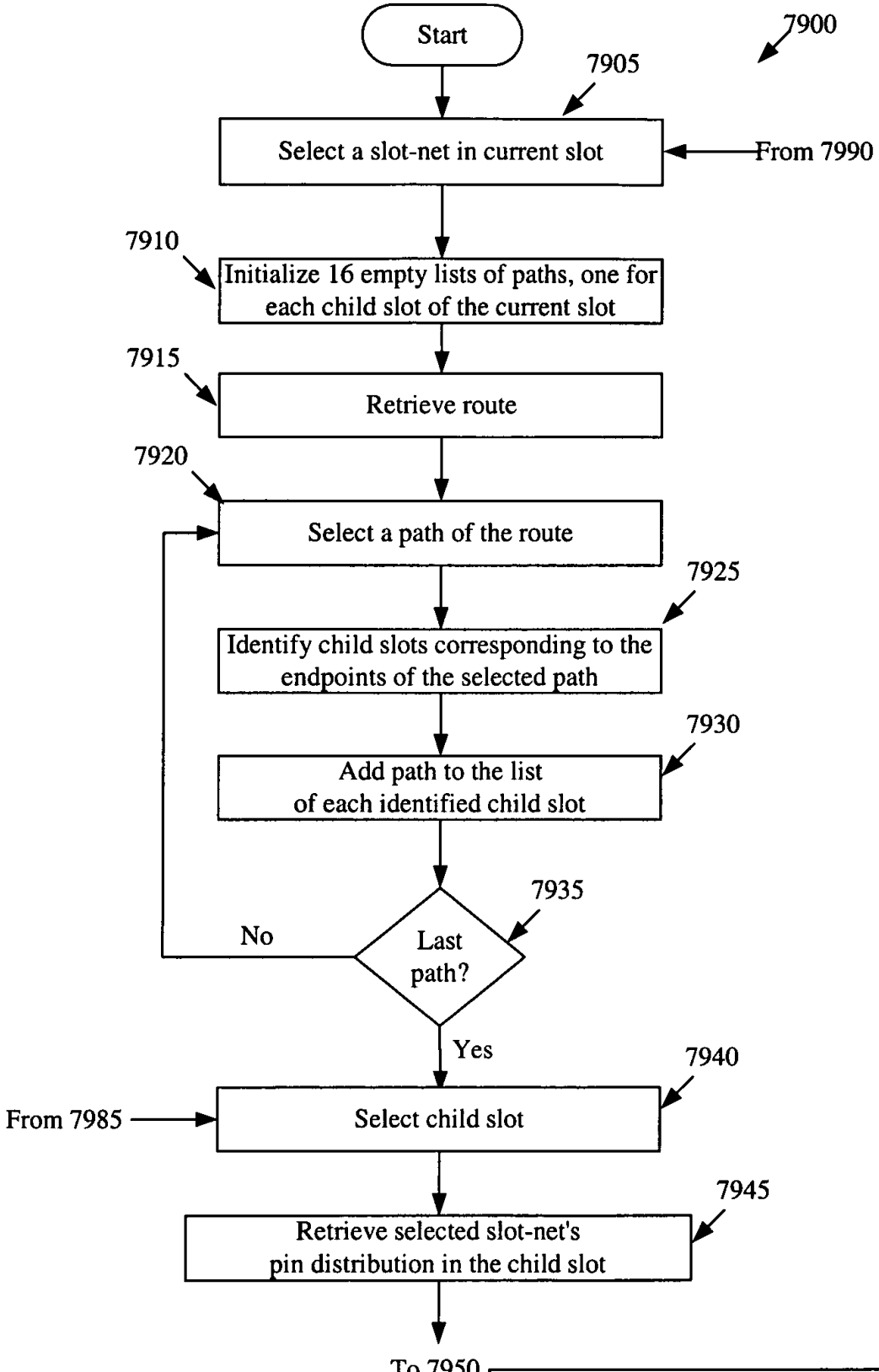

Figure 79A

Figure 79: Figure 79A / Figure 79B

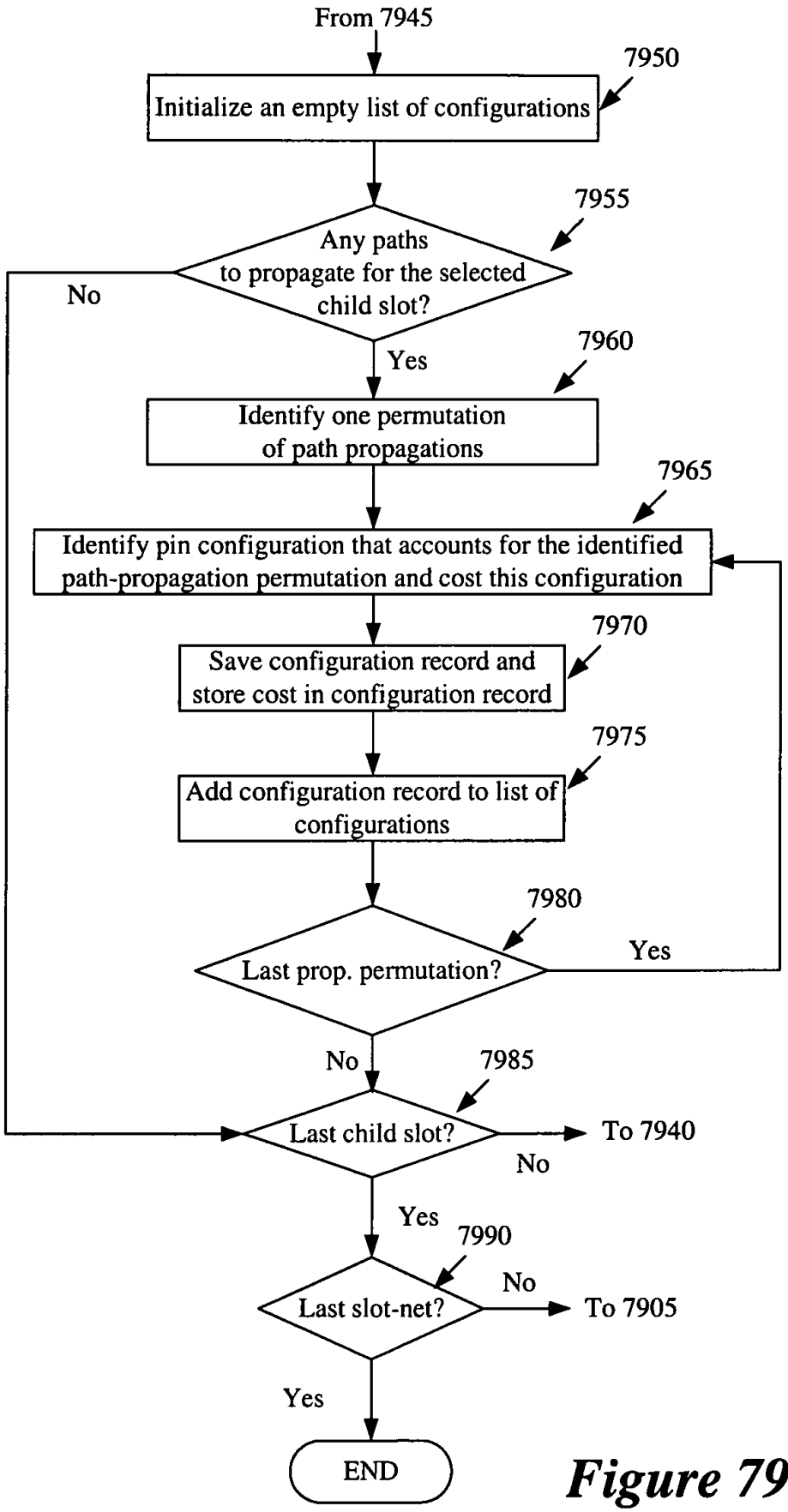
From 7945
7950
Initialize an empty list of configurations
7955
Any paths to propagate for the selected child slot?
No
Yes
7960
Identify one permutation of path propagations
7965
Identify pin configuration that accounts for the identified path-propagation permutation and cost this configuration
7970
Save configuration record and store cost in configuration record
7975
Add configuration record to list of configurations
7980
Last prop. permutation?
Yes
No
7985
Last child slot?
To 7940
No
Yes
7990
Last slot-net?
No
To 7905
Yes
END

Figure 79B

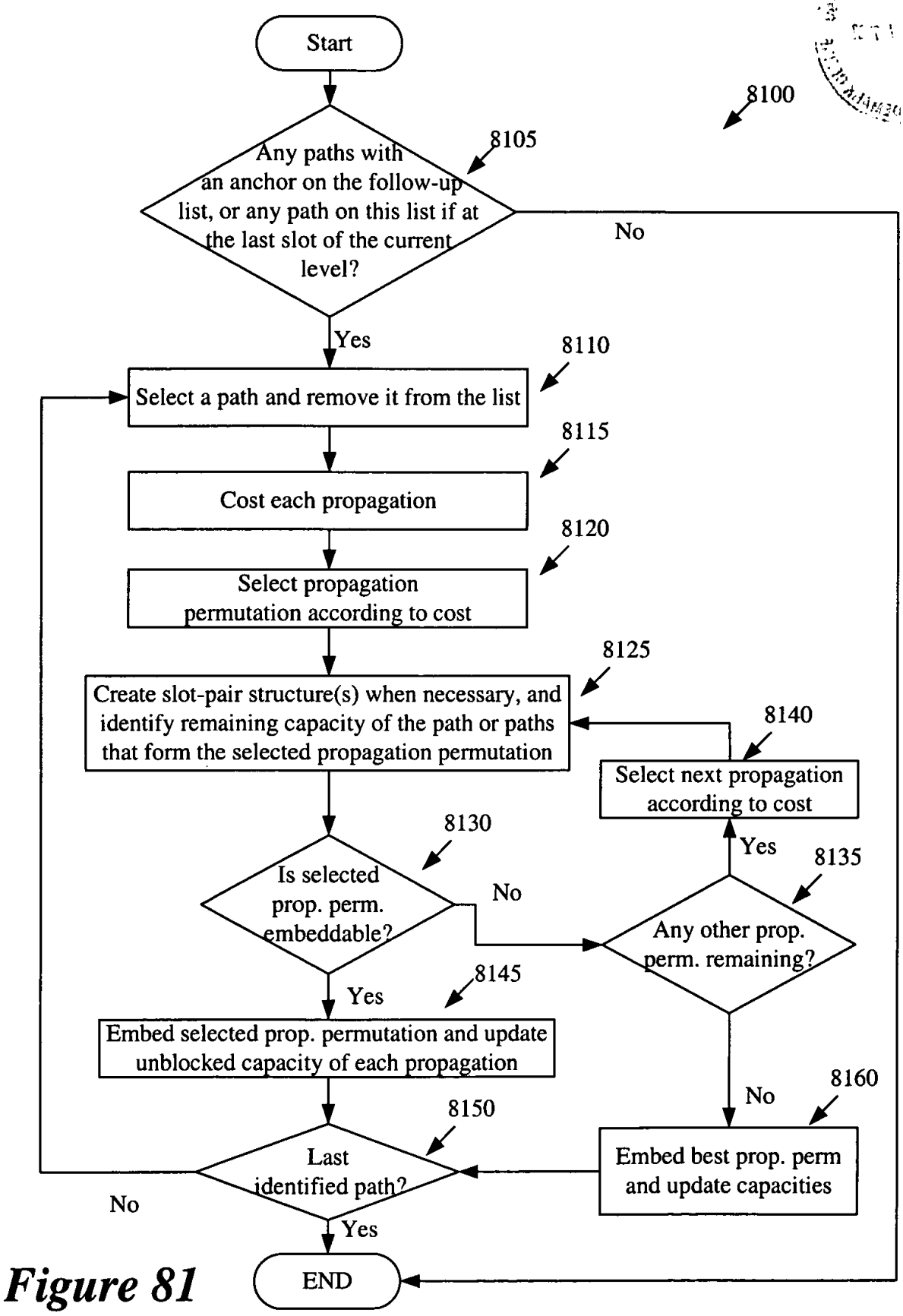
Start
8100
Any paths with an anchor on the follow-up list, or any path on this list if at the last slot of the current level?
8105
No
Yes
8110
Select a path and remove it from the list
8115
Cost each propagation
8120
Select propagation permutation according to cost
8125
Create slot-pair structure(s) when necessary, and identify remaining capacity of the path or paths that form the selected propagation permutation
8140
Select next propagation according to cost
Yes
8135
Any other prop. perm. remaining?
No
8130
Is selected prop. perm. embeddable?
Yes
8145
Embed selected prop. permutation and update unblocked capacity of each propagation
No
8160
Embed best prop. perm and update capacities
8150
Last identified path?
No
Yes
END

Figure 81

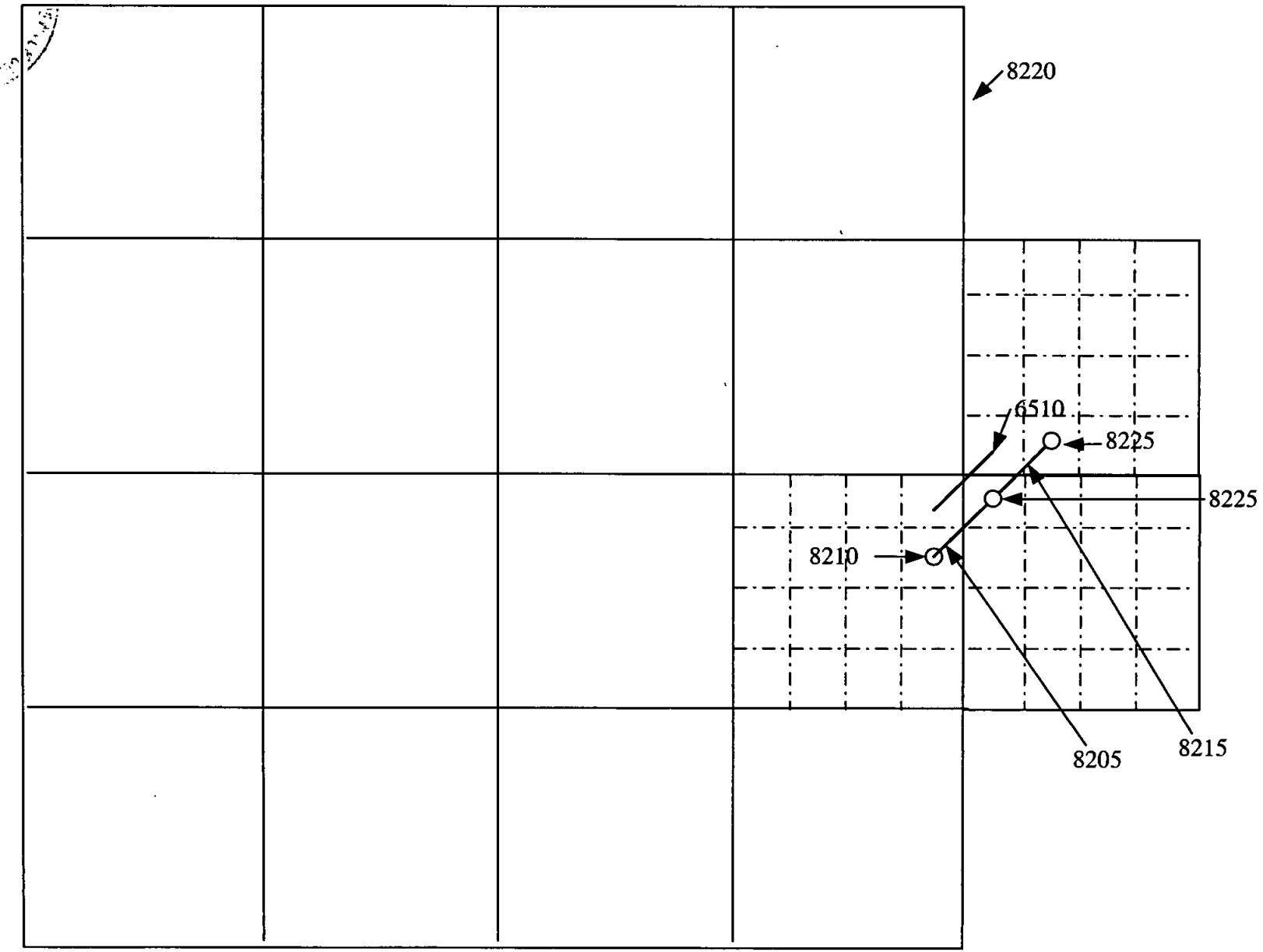

Figure 82

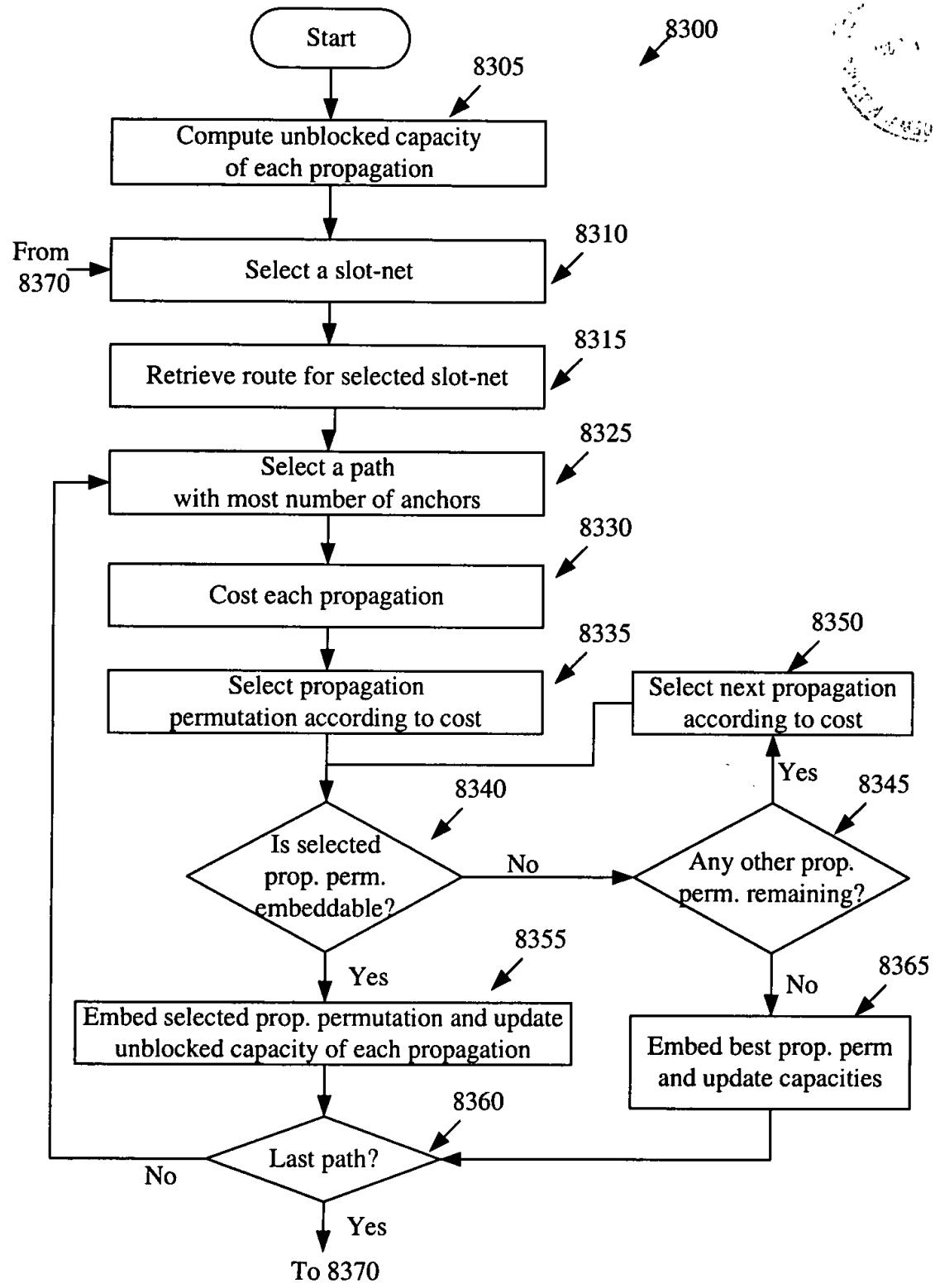

Figure 83A

Figure 83: Figure 83A / Figure 83B

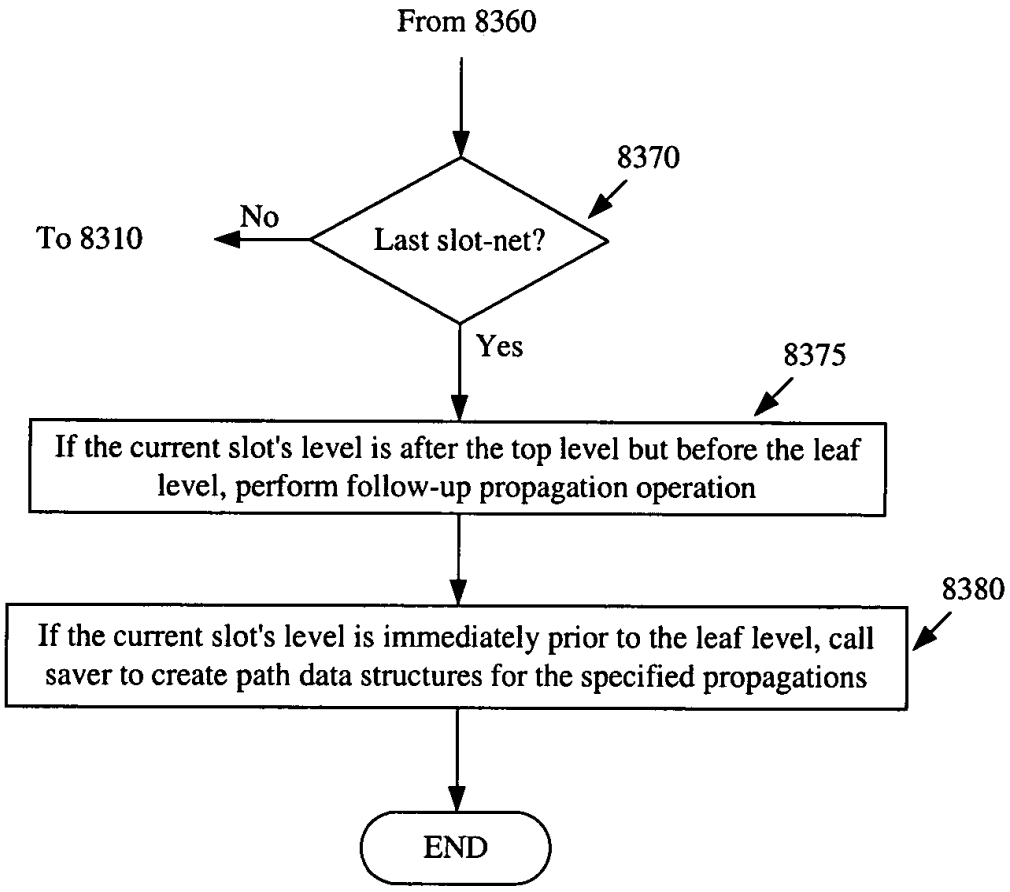
From 8360
8370
No
To 8310
Last slot-net?
Yes
8375
If the current slot's level is after the top level but before the leaf level, perform follow-up propagation operation
8380
If the current slot's level is immediately prior to the leaf level, call saver to create path data structures for the specified propagations
END

Figure 83B

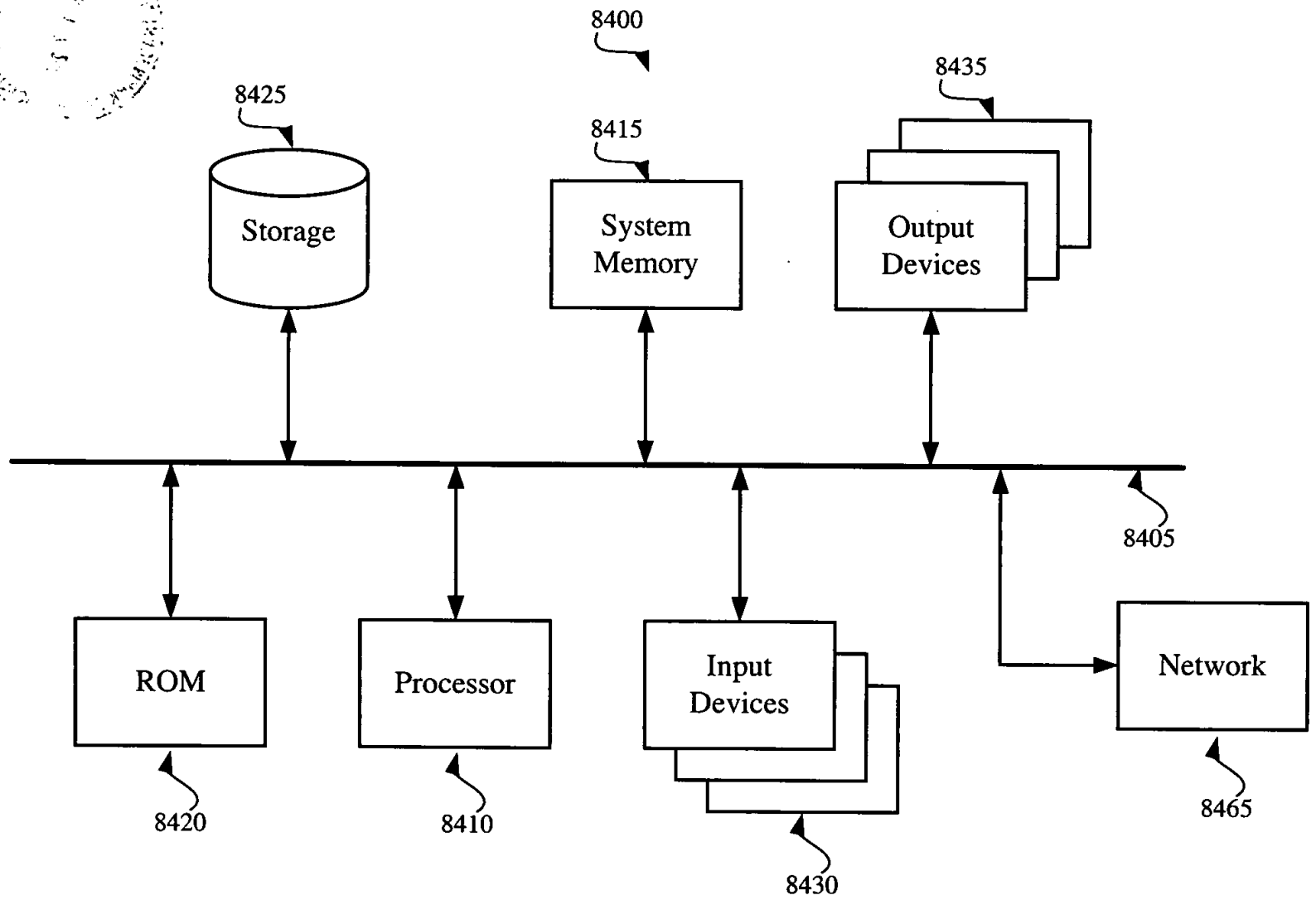
8400
8425
8415
8435
Storage
System Memory
Output Devices
8405
ROM
Processor
Input Devices
Network
8420
8410
8430
8465

Figure 84